DOCUMENT RESUME

ED 238 152 EA 016 247

AUTHOR Douziech, Richard; And Others
TITLE Computer Literacy Activities for Elementary and Middle School Students.
INSTITUTION International Council for Computers in Education, Eugene, Oreg.
PUB DATE 83
NOTE 56p.
AVAILABLE FROM Publications, International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403 ($5.00 prepaid; quantity discounts; add $2.50 for processing on non-prepaid orders).
PUB TYPE Guides - Classroom Use - Guides (For Teachers) (052)

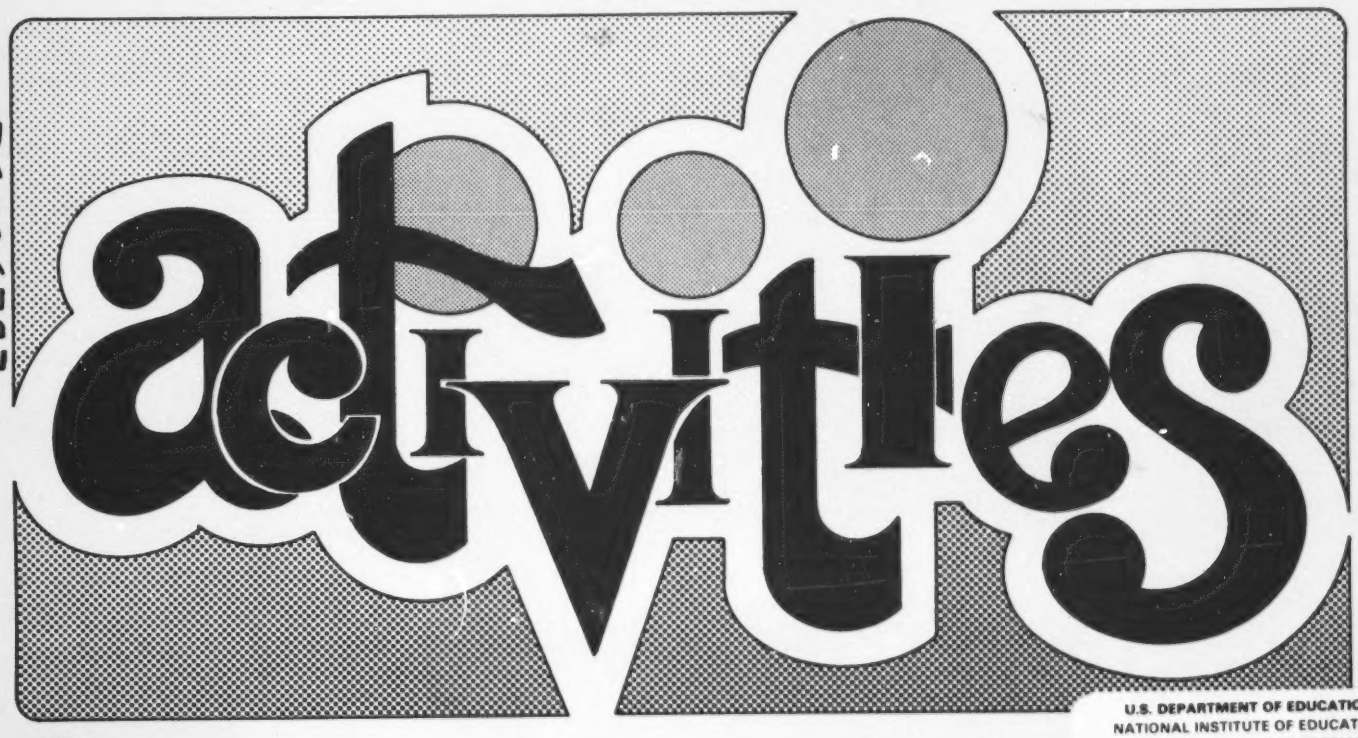
EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Artificial Intelligence; *Computer Literacy; *Computer Oriented Programs; *Computers; Educational Technology; Elementary Education; Instructional Innovation; Instructional Materials; Resource Materials; Teaching Guides

ABSTRACT

All elementary and middle school students can gain five different types of knowledge about computers: (1) how to operate a computer; (2) roles computers play in different subject areas and outside of education; (3) how to direct a computer to aid them in writing, information retrieval, music composition, and creative art; (4) algorithmic thinking; and (5) computer vocabulary. This booklet includes a brief introduction to computers, followed by a collection of 12 activities for a variety of grade levels, designed to help students and their teachers learn more about computers and computer-related ideas. Few activities require access to computers and all can be used by a teacher who is just starting to learn about computers. A glossary and a brief list of additional sources of information are included. (TE)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

Computer Literacy



for Elementary and Middle School Students

ED238152

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

D. Moursund

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

EA 016 247

ICCE Publications

$5.00

About the Authors

The authors of this booklet are all deeply involved in the instructional use of computers at the elementary and middle school levels. Some are elementary or middle school classroom teachers, while others are computers-in-education specialists or media specialists. Together, they possess a tremendous depth and breadth of experience in introducing computer literacy concepts to young students.

This booklet is designed to be used by elementary and middle school teachers who are just getting started with instructional use of computers. Only one of the activities is a "hands-on" experience with computers, and any teacher who has been through a hands-on workshop can supervise students in this initial computer experience. The remainder of the activities are designed to expose students (and their teachers) to fundamental computer-related ideas. These ideas are an essential part of computer literacy and provide a solid foundation for students who will eventually (or concurrently) have substantial experience with computers.

Richard Douziech
Jim English
Sharon Franklin
Ann Lathrop
Jim McCauley
David Moursund
Dick Mowe
Jeff Whittaker

Edited by Anita Best, Associate Editor, International Council for Computers in Education.

For multiple discount prices write:
International Council for Computers in Education
University of Oregon
1787 Agate St.
Eugene, OR 97403 U.S.A.

Computer Literacy Activities for Elementary and Middle School Students

Copyright © ICCE 1983
(First Printing, October 1983)

Preface 3

Introduction 5

Activities

1. Getting Started 9
2. Computers in Our Society 11
3. Small Is Beautiful 13
4. Peanut Butter and Jelly 16
5. Lots of Cards 18
6. GRAPHO (Six Statement Programming Language) 23
7. Acting Out the Computer 28
8. Categories 35
9. Computer Literacy Reference Search 38
10. Card-Section Graphics 41
11. Sorting 44
12. Growing a Binary Tree 47

Appendices

1. Glossary 51
2. Resources 55

Preface

Today's elementary school teacher is expected to be a jack-of-all-trades. Art and language arts, music and mathematics, physical education and science, health education and history, career education and geography—you name it, the elementary school teacher is expected to teach it!

And now we have computers. Computers have the capability of drastically changing both the process and the content of education. They can alter how students learn and what students need to learn. So once again we turn to teachers. Somehow, teachers must now learn to cope with computers and how to introduce computer literacy ideas to their students.

There are three major goals for computers in precollege instruction:

1. All students should develop a useful-tool level (a functional level) of computer literacy.
2. Teaching using computers should occur when it is educationally and economically sound.
3. Curriculum content in all disciplines should adequately reflect the capabilities and limitations of computers.

These are formidable goals for a school system that may have little computer equipment, few computer-oriented resource materials, and few teachers who are themselves functionally computer literate.

But in time computer equipment will become readily available, and computer-oriented resource materials will be developed. Mass production and mass marketing guarantee that this will happen.

Unfortunately, computer literate teachers cannot be mass-produced. The acquisition of computer literacy is an individual activity, highly dependent upon the interests and teaching activities of each teacher. This booklet is for those elementary and middle school teachers who want to begin to develop an educationally-oriented, functional level of computer literacy.

This is a learn-by-doing booklet. It assumes that the teacher has had an initial hands-on workshop and is interested in learning more about computers. This booklet begins with a brief introduction to the field of computers in education. Then, the major part of the booklet is a collection of activities designed to be used at a variety of grade levels. Each activity is designed to help students and their teachers learn more about computers and computer-related ideas. Few require access to computers, and all can be used by a teacher who is just starting to learn about computers.

Once you begin working with computers and computer literacy ideas in your classroom, it is likely you will want to learn more. A glossary and a brief list of additional sources of information have been included to help you.

Introduction

Microcomputer prices continue to decrease while their capabilities continue to increase. It is no longer uncommon to find a microcomputer in an elementary school, and the great majority of secondary schools now make use of computers for instructional purposes. There are about ten times as many computers in people's homes as there are in schools, and computers are used throughout business, government and research.

Out of this rapid proliferation of computers has grown an increasing acceptance among educators that all students should become functionally computer literate. Even preschool-age children can learn to use a computer and can benefit from this experience. Computers can be a useful and important part of young students' lives as they progress through our school system.

Students will learn about computers if teachers and the overall school system provide reasonable assistance and encouragement. Teachers need to integrate computer-related activities into the curriculum and make learning about computers relevant to their students.

Many educators have been led to believe that the only way students can become computer literate is through substantial and regular hands-on experience with computers. This is only partially correct. Experience in using a computer is important, but there are many components of computer literacy that can be learned without the use of a computer. The first activity in this booklet requires that students have access to computers. The remainder of the activities are designed to increase computer literacy (both yours and your students) but do not require use of a computer.

Here are some questions that elementary and middle school teachers frequently ask about computers.

What Is Computer Literacy?

When people first began talking about computer literacy during the late 1960s, they generally meant an awareness level of knowledge. At that time computers were too expensive for students to have extensive hands-on experience. Indeed, many colleges and universities still had quite limited computer facilities. Gradually this situation has changed, so that now the term computer literacy refers to a functional, useful-tool level of computer knowledge and experience.

Elementary and middle school students can gain five different types of knowledge about computers:

1. All students can learn to operate a computer. They can learn to turn on a computer, select a program from a program library, load the program into the computer, run the program and interact with the computer system as the program runs. Even preschool students can gain this functional level of computer knowledge.
2. All students can learn how computers relate to a variety of subject areas and various roles that computers play outside of education. In every academic discipline computers are both an aid to problem solving and a potential source of problems. As students study the conventional methods of problem solving, they can also be learning computer-aided methods for solving the same problems. As students prepare for various careers or other adult roles in our society, they can learn how computers are affecting those careers or adult roles.
3. All students can learn to direct a computer to aid them in writing (through using word processing), information retrieval, music composition and in creative art. If appropriate computer facilities are available, all students can learn to write simple programs in a language such as Logo. In all of these examples the goal is to have the student be in charge of the computer—directing the computer to perform activities of interest and value to the student. In many of these examples the student uses the computer as an aid to exploring non-computer fields such as writing, music, art or geometry.
4. All students can learn about algorithmic processes and algorithmic thinking. A computer is designed to carry out an algorithm—a detailed step-by-step set of directions designed to solve a certain type of problem. The development of an algorithm requires careful, detailed planning and thinking; we call this algorithmic thinking. Even elementary school students can begin to learn to write such detailed sets of directions. They can begin to

learn what types of problems computers can help solve and the type of thinking processes that go into developing computer programs.

5. All students can learn some computer-oriented vocabulary. They can learn general capabilities and applications of computers. In so doing, they are acquiring a foundation of computer-oriented knowledge upon which they can continue to build.

How are computers used in elementary and middle schools?

While there are literally dozens of ways to use computers in a school curriculum, the following general categories cover most of the most important applications.

1. Teaching (learning) using computers.

There is now substantial research evidence to support the contention that in a variety of subjects computers can help most students to learn faster. Certainly computer-assisted learning (CAL) is of growing importance to educators and students. Twenty years from now CAL may well be the dominant mode of instruction in schools in the United States.

2. Computer-managed instruction.

Computers can keep detailed records of student performance; they can grade certain types of homework and tests; they can help guide students through a well-defined curriculum. In all of these activities a computer helps to relieve the "busy work" and classroom management burden faced by teachers.

3. Teaching (learning) about computers.

If students are to gain a meaningful, functional level of computer literacy, they need to have substantial hands-on experience. They need to learn the capabilities and applications of computers in a learn-by-doing environment. Equally important, students need to learn something about the underlying ideas in computer science such as algorithmic processes and algorithmic thinking.

4. Computers as an aid to problem solving.

Over the long run problem solving, defined in a very broad sense, will prove to be one of the biggest categories of educational computer use. Suppose, for example, that a student is faced by the problem of writing an essay. A computerized information retrieval system, a word processor and a computerized spelling checker are all useful aids. With such aids the student can focus more upon content and organization, spending less time on the mechanics of writing and rewriting.

In math, science or social science a student may be faced by a large collection of data. Perhaps a bar graph or a pie chart would help solve the problem of interpreting the data. An appropriately programmed computer can produce a graph in seconds, freeing the student to focus upon higher level, creative processes such as making effective use of a graph to analyze a collection of data.

How can I learn more about computers?

You already know a great deal about computers. You have gained knowledge of computers through reading, watching television, going to movies and talking to people. Computers have been readily available and widely used in our society for about twenty years. Just by living in this society you have learned a lot about computers; in the same manner, you will continue to learn more in the future.

You may not be consciously aware of some of your computer knowledge. For example, our telephone system is heavily computerized. When you use a telephone you are making use of computers that route calls, bill for services and monitor the overall functioning of the telephone system. Many of the newspapers and magazines you read are produced using computerized typesetting equipment. Computers may print your paycheck and send withholding data to the Internal Revenue Service.

This booklet is but one of many teacher-oriented publications of the International Council for Computers in Education. After you finish reading this booklet you will want to seek out other sources of information. The Resources section at the end of this booklet will help you.

The central theme of this booklet is "learn by doing." As a classroom teacher, you have an excellent and exciting opportunity. What you want to learn is how to help your students learn more about computers. Like you, they already have some knowledge. You can work with them, exploring some of the activities given in this booklet. You and your students will learn together.

Learning is often hard, and it helps to have some assistance. Get one of your fellow teachers to work with you as you read this booklet and try out its ideas. Or, seek out a teacher who is already making use of computers. Such teachers usually enjoy helping other teachers learn about computers.

Finally, think about attending a computer conference or taking a workshop or longer course. Most states now have a computers-in-education organization of teachers. These organizations usually publish newsletters and hold annual conferences. Many colleges of education offer computers-in-education courses for teachers. There are summer courses in every part of the country.

How can I get started in helping my students become more computer literate?

It is important to realize that helping your students become more computer literate is very important—both to you and to your students.

There is no "best" first thing to try with your students. A science film that you are showing for other purposes, but which just happens to mention computers, might provide a good starting point. A popular movie or a trip to a science museum may provide a starting point. An interesting way to get started is to ask your students what they know about computers. Get a classroom discussion started. It is likely that the parents of some of your students own computers or work with them. All of your students have seen computers on television. Ask them what they think computers can do and what they think that computers cannot do. You may be surprised by their responses.

Some of your students will probably mention video arcade games. Find out how many of your students have played these games and what they like best about them.

If you and your students decide that it would be fun to spend more class time learning about computers, you will need to do some serious planning. A key to what you can accomplish in class is the amount of computer access you can arrange for your students. It is essential that you arrange some access. Perhaps all that you will be able to arrange is a field trip. More likely, you will be able to arrange to have a computer brought to your classroom and for all of your students to get some hands-on experience.

Along with arranging computer access and hands-on experience, you will want to set some goals and work out some lesson plans. The goals you set should be based upon your knowledge of computers and how much computer access you are able to arrange. A set of goals should take into consideration the five parts of computer literacy discussed earlier in this section. A sample set of goals is given below.

At the end of the computer unit of study every student will:

1. Use computer vocabulary correctly, both orally and in writing. (You might specify a list of 10-20 words.)
2. Know how to gain access to a computer system, select, load and run a library program.
3. Learn to follow an English language "computer program-like" procedure to accomplish a simple task.
4. Learn to write an English language procedure to specify how to solve some type of problem or accomplish some task.

You will notice that none of these goals involves writing computer programs. It is not appropriate that you attempt to teach your students to program if you don't know how to program. Moreover, serious instruction and practice in programming requires substantial time and access to computers. Such computer access and class time will likely be commonplace five to ten years from now, but currently it is unusual.

Once you have worked out some goals and corresponding lesson plans, you are ready to begin. It will probably not go smoothly, and you will have some nervous moments. If a particular lesson doesn't go as expected, face this fact openly and discuss it with your students. Let them know that you are learning along with them, and let them see how you cope with a new learning situation. Above all, HAVE FUN!

Activity #1

GETTING STARTED

SUBJECT AREA: Computer Literacy.

SKILLS TAUGHT: Computer-oriented vocabulary and introductory skills in computer use.

GROUP SIZE: Initially, the whole class. After the introductory phase, students will work individually or in groups of two or three with a computer.

FACILITIES AND MATERIALS: At least one microcomputer, preferably with a large display screen. Demonstration software (this is discussed in the Procedure section). Chart paper and felt pens or a chalkboard and chalk.

OBJECTIVE: To provide all students with a common core of computer vocabulary and hands-on computer experience. Students will demonstrate their understanding and recall by writing and/or discussing what they have experienced and learned.

PROCEDURE:

This activity is directed towards two of the most important aspects of computer literacy. Students will begin to develop a functional level of computer knowledge, while improving their knowledge of computer-related vocabulary at the same time. The teacher can build upon this foundation in future activities.

For this activity it is essential that you obtain the use of a microcomputer in your classroom for several days. If your school doesn't own one, try to borrow one or more from:

1. Another school or school district. (Many school districts now have a "pool" of machines that can be loaned out to individual schools.)
2. A parent of one of your students, or from a fellow teacher. A surprising number of teachers now own computers.
3. A computer store.

As a last resort, arrange a field trip to a school that has a number of microcomputers. Have each of your students gain some hands-on experience during this trip.

In planning this activity you will want to select two or three sample programs that adequately demonstrate the capabilities of a microcomputer. For example, select a program that illustrates the effective use of computer graphics, a computer simulation, a drill and practice program, and a program that makes effective use of music output. (One program may have several of these features.) Run each of these programs yourself. Imagine your students' reactions to running these programs.

The planning phase discussed above assumes that you have been in a computer workshop and/or have had some other hands-on experience with computers. A half-day or one-day workshop is sufficient if it is specifically designed to fit your needs.

Once you have completed initial planning, you can introduce the computer to your class. It is essential to your own learning that you do this yourself, rather than have an outside "expert" come into your class to introduce the computer. Every teacher can easily learn how to turn on a microcomputer, load a program and run a program. Since these are things you want your students to learn, you should be able to model the desired behavior.

A good way to begin is by exploring the computer knowledge of your class. One way to do this is by having the class generate a list of computer-oriented vocabulary words. You

might want to divide the class into small groups, giving each group chart paper and a felt pen. The goal of each group is to write down as many computer-related words as possible.

Combine the students' words into a large list. Discuss the meaning of some of the terms, seeking out students who can define them. Indicate to the students that these terms will be used in spelling and vocabulary exercises to help them learn more about computers.

Demonstrate the microcomputer by showing how to turn it on, loading and running a program and interacting with a program as it runs. Then select a student volunteer to do the same task. Once one or two students show that they can do what you want done, the rest of the students will accept the task as being possible.

The goal is to have every student learn to turn on the machine, load and run a program, and interact with a program as it runs. Some of your students probably will be quite skilled at this task. You may want to use them as aides. Since it is desirable that every student get at least 15 to 30 minutes of hands-on experience, this activity will extend over several days. The aides can provide individualized help to students who have trouble getting started with computers.

After every student has used the computer, give the class the assignment of writing a letter to a parent, grandparent or friend, describing the experience. You might suggest that they write about what they liked and didn't like, and what else they would like to learn about computers.

EXTENSIONS:

Each Activity in this booklet is followed by an Extension section containing some related activities.

1. As an on-going class activity, create a bulletin board display. Encourage your students to bring in pictures, newspaper articles, advertisements and other computer-related materials.
2. Organize a survey of the calculators, electronic video games and computers in your students' homes. Have each student determine how many of these objects are available where they live. Help the class to organize summary statistics on the results.
3. Have your students talk to their parents, siblings and other adults about computers. Each student should try to find an adult who makes use of computers on the job or at home. The student should write a short report on how that adult uses a computer.
4. Have your students talk to their parents or other adults to determine their attitudes about computers. Do they feel that they understand computers and can cope with computers? Do they want their children to learn about and use computers? Students may want to compare the attitudes about computers of someone who works with computers to someone who does not. Students should be asked to write short reports on these interactions.

Note: The eleven remaining activities presented in this booklet do not require access to a computer. These activities can be used at various grade levels and in any order. It is important that your students relate these activities to their initial hands-on computer experience. You can help through the discussion used to introduce each activity. If possible, try to give your students additional experience in using a computer. Even fifteen minutes of computer usage per student per week will serve to make the computer literacy activities more concrete and relevant.

Activity #2

COMPUTERS IN OUR SOCIETY

SUBJECT AREA: Social Studies.

SKILLS TAUGHT: Divergent thinking, awareness of the impact of technology upon our society.

TIME: One hour per day for two days.

GROUP SIZE: Whole class, divided into groups of three to four students.

FACILITIES AND MATERIALS: A large sheet of chart paper and felt pen for each group.

OBJECTIVE: To help students consciously become more aware of technology and its effect upon our society. Students will be able to discuss some roles of computers and other technology in our society, giving applications from education, business and the home.

BACKGROUND:

Computers are but one aspect of our high technological society. Our dependence upon this technology continues to increase rapidly. A computer literate student has a high level of awareness of computer technology and its effect upon people. The student relates this to other technology and the overall impact of technology upon our society.

PROCEDURE:

Day 1: Introduce the idea that throughout history people have created tools and machines to make life easier and more pleasant. Many of these tools and machines perform functions that are similar to those of our bodies, but with greater speed, strength or accuracy. For example, a bicycle extends the capabilities of our legs while a telescope or microscope extends the capabilities of our eyes. A club extends or supplements the strength of one's arm.

To increase student awareness of this concept, divide the class into groups of three or four. Provide each group with chart paper and a felt pen. Each group is to make a list of tools and machines that extend human physical capabilities. You might suggest that they break the task into smaller tasks: find extensions of arms, legs, eyes, ears, teeth, voice and sense of smell or taste. (Note that breaking a large task into smaller tasks is a key concept of problem solving and is a key concept in computer programming.) Encourage each group to make use of encyclopedias and other reference books. (Building upon the work of others in using reference books is another key concept in problem solving and in using computers to solve problems.)

Allow each group to work for 20 to 30 minutes, or until progress ceases. Then have groups share their ideas. You might encourage discussion by asking questions such as, "What is the oldest tool?", "What is the newest tool?", "What is the least expensive or most expensive tool?", "What is the most commonly used tool?", "Which tools have been in use for more than 100 years?"

The emphasis in the first day is upon extensions to the physical capabilities of humans. The second day focuses upon extensions to mental capabilities. Such extensions are apt to be listed in the first day lists. You can suggest that they be saved for the next day.

Day 2: Explain to the class that they are to continue the activity of the first day, but this time they are to look for extensions to the capabilities of the human brain. As before, divide the class into groups and provide each group with chart paper and a felt pen. They will probably list aids such as calculators, books, robots, computers, watches and so on. The more divergent the thinking becomes, the better.

There is no clear or easy distinction between aids to the physical body and aids to the brain. What is important for students to realize is that they are surrounded by aids—that these aids are an integral part of their environment. Reading, writing and arithmetic are also aids, but they are substantially different from the telescope, hearing aid or hammer.

After groups complete their lists, you will want to encourage sharing and discussion. In what sense is a calculator an aid to the brain? Is this similar to a watch?

Almost all students have used a calculator at home or at school. Many hold the opinion that using a calculator to do arithmetic is "cheating." You should encourage this idea to be raised, because it is an excellent idea for discussion. It isn't cheating to use a hammer to drive a nail or a saw to cut a board. It isn't cheating to use a telephone to talk to a person located miles away. Almost all adults make use of calculators. In what sense is it cheating to use a calculator to do arithmetic in school?

The direction of the discussion should be toward computers and their use as an aid to mental processes. A computer can help a person to retrieve information from a library. It can compose or help to compose music. It can solve math problems or gather data in a scientific experiment. All of these things can also be done by hand or with the aid of pencil and paper.

An interesting historical note can be introduced by considering initial (historical) reactions to such items as the telephone or ball point pen. New technology always meets with resistance, and the resistance may be well justified. The technology represents potential change, and often one cannot tell in advance if the change will be for the better.

There are a number of assignments that you might give your students. For example, you might ask them to write about or talk about what life would be like if we did not have electricity. Or, you might have them talk to some older people to find out what it was like to live in a world without television.

EXTENSIONS:

1. Create a bulletin board display of aids to the physical body and aids to the brain. Have students bring pictures from home.
2. Have each student select one aid and write a report on its effect upon our society.
3. Discuss the concept of automation with the class. Help them to become aware that automation existed long before computers, that automation increases worker productivity and that automation decreases the number of people needed to accomplish a given task. Specific assignments might include discussing automation and related employment ideas with adults and reporting what they say.
4. Find a film on machines in business and industry, perhaps one emphasizing automation. Use the film to stimulate class discussion on how machines change people's jobs and the type of education needed to get a good job.
5. Ask students to name television, movie and comic book characters that are machine-like or computer-like. For example, Steve Austin and Jamie Summers are "bionic." Your students will probably be aware of real world artificial body components such as artificial legs, artificial hearts and kidney machines. Help your students to recognize differences between the make believe world of television and comic books and the real world.
6. A teacher, a book and a computer can each be an aid to learning. Have each of your students make use of computer-assisted instruction materials to help them learn a new topic. Then have a class discussion about learning from other people, learning from books and learning from a machine.

Activity #3

SMALL IS BEAUTIFUL

SUBJECT AREA: Social Studies, Consumer Education.

SKILLS TAUGHT: Divergent thinking, awareness of impact of technology on our society.

TIME: One hour per day for two days.

GROUP SIZE: Whole class, divided into groups of three or four.

FACILITIES AND MATERIALS: Large sheets of chart paper and felt pens. Pictures, slides or movies showing early computers and more modern microcomputers. (Your school district's film library probably has some old computer films. Check a recent film catalog for more modern materials.)

OBJECTIVE: To help students become aware of technological change and the effects of technology on our everyday lives. Students will study electric motors, noting their wide use. They will see the parallel between this example and computers, as small computers (microcomputers) are becoming commonplace.

BACKGROUND:

Most people are not yet aware of how rapidly microcomputers and microprocessor-based equipment are coming into general use. Millions of general-purpose microcomputers are now being manufactured and sold each year in the United States. Billions of microprocessors ("a computer on a chip") are now in everyday use. An important part of computer literacy is recognizing where and how computers and computerized equipment are being used. Generally speaking, computers are being used to help solve problems that can also be solved without the use of computers. Computers are used to produce equipment that may be cheaper, more reliable, more versatile or more useful.

PROCEDURE:

Part I: Small Electric Motors

The industrial revolution in the United States and elsewhere depended upon developing sources of power to run machinery. Water power, steam power and later, electrical power supplemented animal power and human muscle. These power sources were large, expensive machines. They led to the building of large factories located near sources of power.

Eventually, the small electric motor was developed. Electricity in the home could power a vacuum cleaner, washing machine, sewing machine or dozens of other machines. The home and life in our society were substantially changed.

1. Discuss with the class the above ideas and how we are literally surrounded by small electric motors. These motors are a relatively recent invention.
2. Divide the class into groups of three or four students. Provide each group with a large sheet of chart paper and a felt pen. Each group is to make a list of household items containing small electric motors. Be sure to include battery-powered motorized items such as toys and electric clocks.

3. Combine the individual group list to create as large a list as possible. Work with the class to make an estimate of the number of small electric motors in a typical house.
4. Help students to think about what would happen if all of these motors were unplugged and all batteries removed. After some class discussion, have students write a short story on what life would be like in their home if there were no electric motors.

Part II: Computers

The first general-purpose electronic digital computer built in the United States was called the ENIAC and was completed in December 1945. It contained 18,000 vacuum tubes, consumed an enormous amount of electricity and filled a large room (about the size of a classroom).

The UNIVAC I was the first mass-produced computer; it came on the market in 1951. While smaller and more reliable than the ENIAC, it still contained thousands of vacuum tubes and was quite large. It cost well over a million dollars, so only large companies and government agencies could afford to purchase one.

The transistor was invented in 1947 and became commonplace in computers in the early 1960s. Transistors are smaller, use less power and don't wear out as fast as vacuum tubes.

A "chip," or integrated circuit, is a fingernail-sized circuit containing thousands of individual transistors. A single chip may contain all of the circuitry needed ir a hand-held calculator.

Continued progress in transistor technology led to the development of the microcomputer in the mid 1970s. Microcomputers have been mass-produced and widely sold since 1977. Such machines are becoming a common item in schools and homes, and they are now used throughout business and government.

In 1981 Hewlett-Packard began producing a chip containing 450,000 transistors. Another company announced a million transistor chip early in 1983. This rapid progress is apt to continue for another ten to twenty years. It will make very powerful computers more readily available than small electric motors.

It is useful to distinguish between general-purpose microcomputers and special-purpose microcomputers. A general-purpose microcomputer has a typewriter-style keyboard and is programmable by its user. It can be programmed to solve a wide variety of problems. The circuitry in an electronic video arcade game may be identical to that in a general-purpose microcomputer, but the video game is a special-purpose machine, designed to perform a specific entertainment task. Special-purpose computers are being built into thermostats, cars, microwave ovens, television tuners, clock radios and many other consumer items.

1. The historical perspective of computers getting smaller, more capable, less expensive and more readily available is an important part of computer literacy. Help your students to understand that electronic technology has changed very rapidly and continues to change rapidly. Give them some examples of computerized devices that are now quite common, and encourage your students to help extend the list.
2. Have each student bring to school pictures of computers and devices containing computers. Ads for computers and computerized devices can be found in most magazines and newspapers. These pictures might be used to form a bulletin board display.

EXTENSIONS:

1. Have students select some non-computer technology and write a paper on its historical development and implementation. Good topics include telephone, radio, television and communication satellites.
2. Have students study the history of computers or a particular person who has been deeply involved with the development of computers. Each student can write a paper or make an oral report to the class.
3. Late in 1982, Casio began selling a wrist watch which contained a 1,711 word Spanish/English dictionary. A wrist watch that can "speak" the time (voice output) has been commercially available for several years. By 1990 we can expect that these technologies will be combined to produce a wrist watch with a several thousand word language translation dictionary and voice output. This type of computer technology could be used to develop many other aids to students. Have each student "invent" a computerized device that they would like to own. What would it be able to do?

4. Portable, battery-powered, general-purpose microcomputers became commercially available in the early 1980s. Many college students and business people carry cassette tape recorders. A portable computer may be as small as a cassette tape recorder. Lead your class in a discussion of possible uses of a portable computer.

Activity #4

PEANUT BUTTER & JELLY

SUBJECT AREA: Language Arts.

SKILLS TAUGHT: Developing and following a written set of directions, precise written expression, critical thinking, algorithmic thinking.

TIME: A minimum of one hour. This activity can be spread (no pun intended) over several days.

GROUP SIZE: Whole class.

FACILITIES AND MATERIALS: Bread, peanut butter, jelly, knife and lots of paper towels.

OBJECTIVE: To introduce students to the idea of a detailed step-by-step set of directions. Students will learn about bugs (errors in a set of directions) and debugging (locating and correcting errors). Students will gain experience in developing and following a detailed set of directions.

BACKGROUND:

A computer program is a detailed step-by-step set of directions. A computer is a machine that can store such a set of directions in its memory and then execute the directions (carry them out) rapidly, accurately and automatically. The essence of computer programming is figuring out how to solve a problem and then specifying a detailed step-by-step set of directions to tell the computer how to solve the problem.

Computer scientists call a detailed step-by-step set of directions a procedure. Mathematicians call it an algorithm if it is designed to solve some type of math problem in a finite number of operations. Algorithmic thinking is the type of thinking one does in developing and testing algorithms or procedures. Computer thinking, algorithmic thinking, procedural thinking, algorithmics and so on are all terms for the same general idea.

A key concept in algorithmic thinking is mentally following an algorithm or procedure, checking its correctness and trying to detect errors. Computer scientists call an error a bug. The detection and correction of errors is called debugging.

This peanut butter and jelly activity is a "classic." It has been widely used and has been written up in many publications. See, for example, the article by Marilyn Burns, "Getting Kids Ready for Computer Thinking," *The Computing Teacher,* Vol. 8, No. 1.

PROCEDURE:

Tell your students that they must write a set of directions for making a peanut butter and jelly sandwich. The directions are to be very detailed, designed to be read and followed by a person who has never made or even seen a peanut butter and jelly sandwich. Each student is to write a set of directions.

As soon as students complete their directions, collect them. Quickly glance through them to find one with some serious flaws. Alternatively, you, the teacher, can write a set of directions that is seriously flawed.

Hand the selected set of directions to a student who is a good reader. The student is to read the instructions one step at a time. You will carry them out. The reader must read precisely what is on the sheet, and you must follow these instructions. You are not allowed to use "com-

mon sense" to fill in gaps or avoid disaster.

For example, suppose that an instruction specifies, "Put the peanut butter on a slice of bread." You could pick up the jar of peanut butter and set it on a slice of bread. Suppose an instruction says, "Take some peanut butter out of the jar and spread it on a slice of bread." You might scoop a handful of peanut butter out of the jar with your hands and spread it on the slice of bread. An instruction, "Now spread jelly on the bread," could be interpreted to spread jelly on both sides of the bread. An instruction, "Set the slice down," could lead you to set it down jelly side down. "Put the two slices of bread together," could be interpreted as putting them together with the peanut butter and jelly on the outside or setting them side by side on a table.

The point is, it is very difficult to write a foolproof set of directions. One must understand precisely what the agent (the person or machine who will follow the directions) is able to understand and to do. A substantial part of learning to program a computer is learning precisely what the computer can do and how it understands and executes a set of directions.

To continue the activity, hand each student a set of directions written by another student. Have each student go through the set of directions mentally, imagining the consequences of following them precisely. Each student is to detect bugs in the set of directions and write suggestions for improvement.

Next, you can divide the class into small groups and have each group work on developing a foolproof set of directions for making a peanut butter and jelly sandwich. After each group finishes, have groups exchange their solutions. The idea is to find bugs (errors, flaws) in solutions proposed by other groups. At this stage of the activity,you might want to have individual students or groups of students accept this as a homework assignment with class discussion to continue the next day.

EXTENSIONS:

1. In the debugging process, students will see the need to write in new instructions between the lines of the original instructions. One way to handle this is to leave blank lines between instructions in the first writing. Another way is to write instructions on 3 x 5 cards or slips of paper and stack them in the desired order. Still another way is to number the original set of directions 10, 20, 30, 40, etc. and then allow new instructions such as "5 Open the jar of peanut butter" to use the in-between numbers. The agent who is to execute the directions, executes them in order of increasing instruction number.

   A programming language such as Logo is designed much like the stack of cards idea. A programmer can insert new instructions between existing instructions, just like inserting a new card. The BASIC programming language uses the instruction numbering idea given above.

2. There are many tasks, not as messy as making a peanut butter and jelly sandwich, that can be assigned to students. For example, "Write a detailed set of directions on how to locate and check out a mystery book from the school library," or "Write a set of directions on how to look up a phone number in a telephone book."

3. Have your students make lists of tasks which can be accomplished by following a detailed step-by-step set of directions. The goal is to help students become more aware that there are many different tasks that can be formulated as relatively routine step-by-step sets of directions. Part of algorithmic thinking is recognizing when a problem can be solved by an algorithm or procedure.

4. Show students a simple computer program written in BASIC or Logo. Follow through the program steps, explaining how each program statement has precisely one meaning to the computer.

Activity #5

LOTS OF CARDS

SUBJECT AREA: Language Arts, Mathematics.

SKILLS TAUGHT: Reading and following directions, counting, organizing and recording data, developing algorithmic thinking.

TIME: A few minutes per learning station; total time depends upon number of learning stations.

GROUP SIZE: Any number; students work individually in learning stations.

FACILITIES AND MATERIALS: A learning station is a table or booth containing an activity to be done by a student. Each learning station contains a deck of playing cards (it's okay if some cards are missing) and a set of directions. Details on the directions are discussed later. Each learning station may also contain a clock or other timing device, to be used in the extensions of this main activity.

OBJECTIVE: To give students practice in reading and following a detailed step-by-step set of directions. Students will learn that computers can execute (carry out) a detailed step-by-step set of directions and that the same problem can be solved in a variety of ways.

BACKGROUND:

Computer programming is a creative, thinking, problem-solving task. It is a combination of art and science, drawing upon one's total training and experience.

Computers can be an aid to problem solving in every academic discipline. Computers can process letters, words and sentences as well as they can process digits, numbers and formulas. While this particular activity involves counting and recording data, which might be considered to be mathematics, similar activities are necessary in solving social science problems and problems from other disciplines.

PROCEDURE:

To prepare students for this activity, present them with the idea that a computer program is a detailed step-by-step set of directions called a procedure. It must be written in a language that a computer can "understand"—that is, interpret and carry out. The types of directions a computer can follow are dependent on the nature of both the computer and other equipment attached to it. A computer attached to a welding machine can serve as part of an automated automobile assembly line. A computer attached to a video disk player can serve as a computer-assisted learning machine.

This activity requires that you set up a number of learning stations. Each learning station requires a "deck" of playing cards. For young students, a deck of 10 to 15 cards is recommended. For older students, 30 to 50 cards will be more appropriate. Each learning station needs a **Learning Station Activity Card.** Five samples are given which you may photocopy or just use for models. As you can see, each sample contains a detailed step-by-step set of directions to be followed by a student.

Each student will also need a **Recording Sheet.** You will need to give students some instruction and practice in using a Recording Sheet before they begin to go through the Learning Stations. A sample Recording Sheet follows the activity card samples.

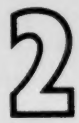
LEARNING STATION ACTIVITY CARD

STATION NUMBER

1. Record the STATION NUMBER on your RECORDING SHEET.
2. This LEARNING STATION contains a deck of playing cards. Put all of the face cards (that is, Jacks, Queens and Kings) into one pile.
3. From the remaining cards, select out all black ones and put them into a second pile.
4. Now count the cards that have not been put into a pile. Record that count on your RECORDING SHEET next to the STATION NUMBER.
5. Put all of the cards back together in one pile and shuffle this pile of cards.
6. You have now completed the activity of this LEARNING STATION.

LEARNING STATION ACTIVITY CARD

STATION NUMBER



1. Record the STATION NUMBER on your RECORDING SHEET.
2. If the number of cards in the LEARNING STATION deck is even, record the word EVEN on your RECORDING SHEET. Otherwise, record the word ODD.
3. Make sure that the deck of cards has been shuffled before you leave this LEARNING STATION. Then you are done with this LEARNING STATION.

LEARNING STATION ACTIVITY CARD

STATION NUMBER

3

1. Record the STATION NUMBER on your RECORDING SHEET.
2. Put all of the spades into one pile.
3. Put all of the hearts into a second pile.
4. Put all of the diamonds into a third pile.
5. Put all of the clubs into a fourth pile.
6. On your RECORDING SHEET record the suit of the pile containing the least number of cards.
7. Now put all of the cards back together into one pile and shuffle it. That completes the activity of this LEARNING STATION.

LEARNING STATION ACTIVITY CARD

STATION NUMBER

4

1. Record the STATION NUMBER on your RECORDING SHEET.
2. Determine the total number of 2's, 3's and 4's in the deck of cards.
3. Record this total on your RECORDING SHEET.
4. Put all of the cards into one pile and shuffle it. You are now through with this LEARNING STATION.

LEARNING STATION ACTIVITY CARD

STATION NUMBER

5

1. Record the STATION NUMBER on your RECORDING SHEET.
2. Put all of the face cards (that is, Jacks, Queens and Kings) into one pile and the rest of the cards into another pile.
3. Count both piles.
4. On your RECORDING SHEET record the **difference** in the number of cards in each pile.
5. Put all of the cards into one pile and shuffle it. You are now through with this LEARNING STATION.

RECORDING SHEET

1. Print your first name here. _____
2. Print your last name here. _____
3. Print today's date here. _____
4. Use the space below to record the results obtained at each LEARNING STATION you visit.

STATION NUMBER	RESULT
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Note: The deck of cards for **Learning Station Number 3** should have one suit with less cards than any other suit. For more advanced students, you should discuss the idea that there may be more than one solution to a problem. You can discuss with them how to handle an activity of this sort when two or more piles contain "the least" number of cards.

If you need more than five learning stations, you should have little trouble creating additional activity cards.

EXTENSIONS:

1. Have a clock visible to a student working in a Learning Station. A student is to record the time when entering and leaving each Learning Station. You will need to provide an appropriately modified Recording Sheet for this. After a student has completed all assigned Learning Station activities, the student is to determine the total amount of time spent in the stations.

2. Many of the results or answers determined by the students are easily obtained in several different ways. One method is by counting, without actually sorting cards into piles. Show students how to do this, and then have them repeat a number of the activities. Note that you can easily modify the correct answers in the stations by adding or deleting a few cards at each station.

   Example: Are there more red cards than black cards in the deck? Count +1 for each red card and −1 for each black card. Do the counting in your head. Suppose that the actual sequence of cards is R B B B R R B R R B R R. The mental counting would go +1,0, −1, −2, −1, 0, −1, 0, +1, 0, +1, +2. The conclusion is that there are two more red cards than black cards.

Activity #6

GRAPHO (SIX STATEMENT PROGRAMMING LANGUAGE)

SUBJECT AREA: Art, Computer Programming.

SKILLS TAUGHT: Computer programming in a six statement, graphics-oriented language.

TIME: A minimum of two thirty-minute periods. This activity can be extended to last for a week or more.

GROUP SIZE: Whole class.

FACILITIES AND MATERIALS: Graph paper with squares of a size appropriate to the manual dexterity of your students. This can be made using a ditto master.

OBJECTIVE: To acquaint students with the rudiments of computer programming in a graphics-oriented programming language. Students will learn a six statement programming language; they will read programs written in this language and write their own programs.

BACKGROUND:

A computer is a machine, designed to follow a detailed step-by-step set of directions. The directions must be written in a language that the machine can "understand"—that is, interpret and carry out. The directions are written by a person—a computer programmer.

A computer programmer must know the details of the programming language to be used and must be skilled in developing directions to be written in this language. The vocabulary of a programming language is generally quite small, so that learning the vocabulary is not difficult. It is thinking creativity, the solving of problems using the language, that is a challenge.

Each computer is designed to be able to interpret and carry out programs written in its machine language. To make programming easier, humans have developed easier-to-learn and more human-like languages such as BASIC, Logo and Pascal. Translation programs have been written to translate from such languages into the machine languages of specific machines. This means that a particular computer may be able to "understand" a number of languages.

Whether a computer understands a particular programming language depends upon the availability of a suitable language translator. Most microcomputers have a BASIC translator built into their circuitry—that is, BASIC is included in the price of the machine. But languages such as Logo and Pascal, if they are available, are extra cost items. A translator for Logo might cost about a hundred dollars for a typical microcomputer. Moreover, Logo requires quite a bit of internal computer memory, so it cannot be run on a computer with a very small amount of internal memory.

Programming languages are designed to fit the needs of specific groups of people. BASIC was designed for a broad range of college students, although it is the most widely used language at the precollege level. Logo was originally designed for elementary school students while Pascal was designed for college-level computer science majors.

A key part of Logo is its graphic, or picture drawing capabilities. The GRAPHO language statements introduced in this activity, FORWARD, BACK, RIGHT, LEFT, PENUP and

PENDOWN, are quite similar to part of the Logo language. GRAPHO is a pencil and paper language, designed for pencil and paper activities. These GRAPHO activities are excellent preparation for a later hands-on introduction to Logo.

There are many advantages to introducing students to GRAPHO before giving them hands-on experience with Logo. All students can participate at the same time (graph paper is cheaper than a classroom set of computers). Students can master some of the Logo vocabulary at their leisure (they can even practice at home). Details of actually dealing with the computer, such as typing in a program, editing, correcting errors and reading error messages, are avoided when using GRAPHO.

PROCEDURE:

The GRAPHO language consists of six statements designed to specify the movement of a pen on a sheet of graph paper. The six statements are FORWARD, BACK, RIGHT, LEFT, PENUP and PENDOWN. To help your students learn these statements, you will want to list them on a chalk board or a piece of chart paper and illustrate their use. The next paragraph explains the meaning of each of these six statements.

Imagine having a felt pen whose tip exactly fills one square of the graph paper. The drawing process begins with the pen poised over a square. PENDOWN places the pen onto the paper, inking in the square. PENUP is used to raise the pen off the paper. In either the PENUP or the PENDOWN position, one can carry out any of four pen movements. When beginning a drawing it is assumed that the pen is "pointed" towards the top of the page. FORWARD 6 means to move the pen six squares in the direction it is pointed. If the pen is in the PENDOWN position, this causes six squares to be inked. PENDOWN followed by FORWARD 6 causes a total of seven squares to be inked. Similarly, BACK 4 moves the pen backwards (opposite to the direction it is pointed) four squares. RIGHT turns the pen 90 degrees clockwise and LEFT turns the pen 90 degrees counterclockwise.

Students using GRAPHO are apt to use a pencil or ball point pen rather than a felt pen whose tip exactly fills a graph paper square. Moreover, you probably want a student to do a number of drawings on one sheet of graph paper. Thus, illustrate the specification of a screen on the graph paper, say a 21 square by 21 square box. At the beginning the pen is poised over the middle square in the 21 by 21 screen. A computer implementation of Logo usually has a screen that is divided into very small dots rather than into squares. A screen might be 250 dots wide by 200 dots high. While this allows more detailed drawings than we can do in GRAPHO using a 21 by 21 screen, the main ideas are the same.

The following program draws a rectangular-shaped box. Notice that we have given the program a name. This makes it more convenient in talking about the program. While we called this particular program BOX, we might just as well have called YELLOW or FRED. A computer does not attach meaning to the name of a program, other than that the name stands for the program.

```
PENDOWN
FORWARD 4
RIGHT
FORWARD 7
RIGHT
FORWARD 4
RIGHT
FORWARD 7
```

The box will be eight squares long and five squares wide. At the end the pen will be down on the paper and pointed towards the left.

In GRAPHO we have called the drawing device a "pen" and talked about the direction the pen is "pointed." In Logo the pen is called a "turtle" and one talks about the direction the turtle is facing. If you have access to Logo for use in your class, you will probably want to refer to the pen as turtle and talk about the direction the turtle is facing.

Now start with a fresh screen (as above, we assume it is a 21 by 21 grid). Figure out the results of executing the following program.

```
DEMO 1
FORWARD 10
LEFT
PENDOWN
FORWARD 10
BACK 20
LEFT
FORWARD 20
RIGHT
PENUP
FORWARD 20
PENDOWN
RIGHT
FORWARD 20
```


24

Figure it out by following through the program steps, drawing on a piece of graph paper. Alternatively, imagine the whole process in your head. It is possible to become quite skilled in "drawing" in one's head. The program draws along three quarters of the edge of the screen. The bottom edge of the screen is left blank.

Suppose we want to add a horizontal line, five squares long and exactly through the center of the screen, to the picture produced by the above program. Two solutions are given below. Follow through each of them. Notice that one must pay careful attention to the direction the pen is pointed.

```
DEMO 2
LEFT
FORWARD 2
PENDOWN
BACK 4
PENUP
FORWARD 2
RIGHT
FORWARD 10
LEFT
PENDOWN
FORWARD 10
BACK 20
LEFT
FORWARD 20
RIGHT
PENUP
FORWARD 20
PENDOWN
RIGHT
FORWARD 20
```

```
DEMO 3
FORWARD 10
LEFT
PENDOWN
FORWARD 10
BACK 20
LEFT
FORWARD 20
RIGHT
PENUP
FORWARD 20
PENDOWN
RIGHT
FORWARD 20
PENUP
BACK 10
RIGHT
FORWARD 8
PENDOWN
FORWARD 4
```

Notice that the two programs are not the same length. There are many ways to write a program that solves a specified problem. Moreover, the shortest program is not necessarily the best program. Ease of reading and understanding and modifying a program are important characteristics.

After working through some examples with your students, hand out several programs to them. This is conveniently done on ditto paper, with the program and a graph paper screen side by side on the paper. A master to photocopy, add a program, and run off is included after the Extensions. Each student is to follow the statements in the programs.

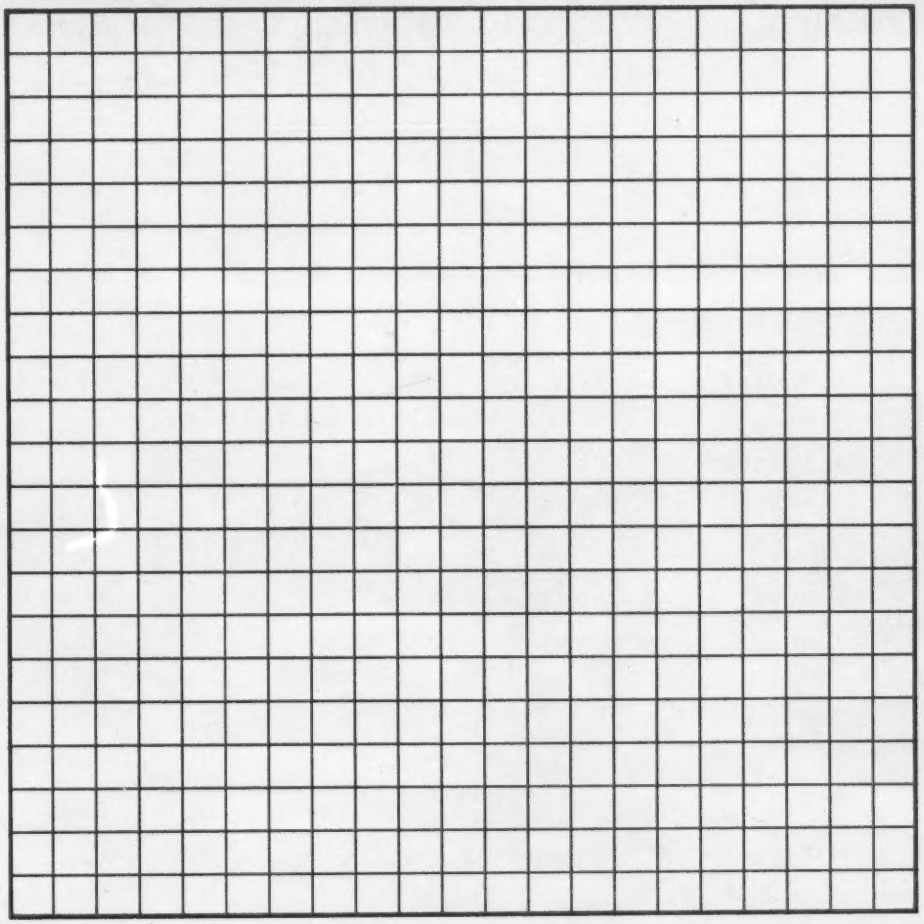
EXTENSIONS:

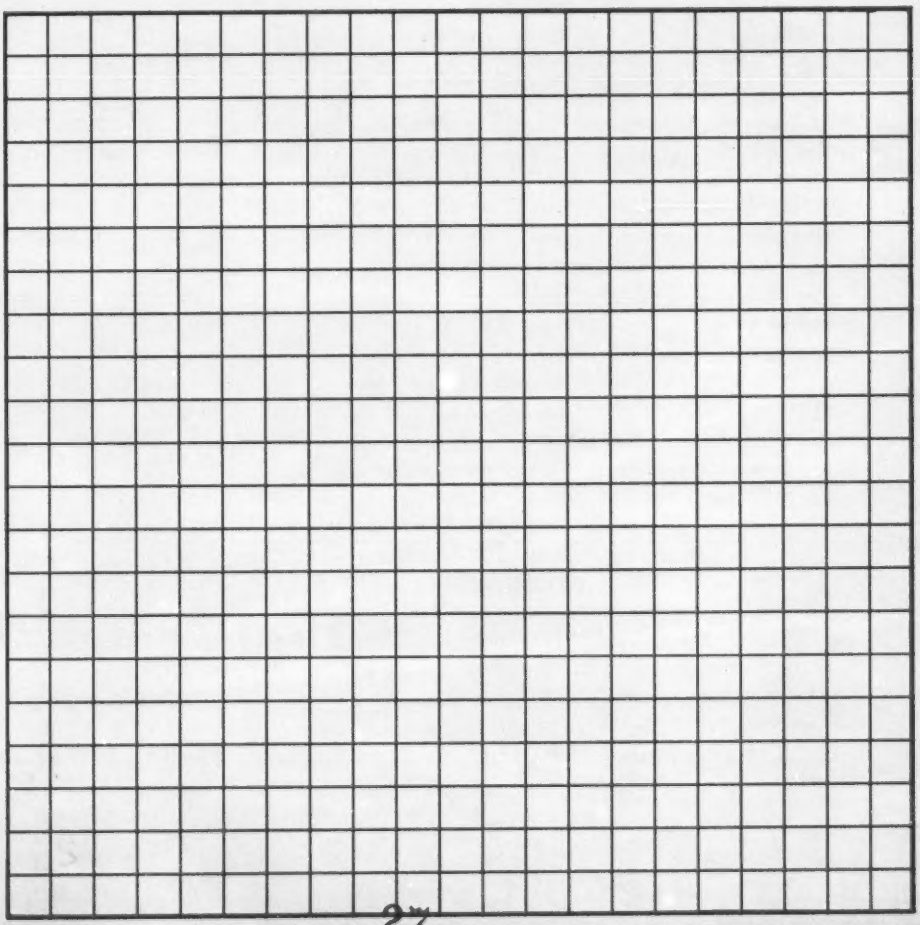
1. Have students write their own programs. For example, you might ask each student to write a program that draws the first letter of their first name, a clown face or a house. After students complete their programs, have them exchange papers and execute each other's programs.
2. Start a "best GRAPHO picture" contest. Entrants must create a program and draw the picture that the program is designed to draw. Award prizes for categories such as "most fun", "most complex", or "most creative". Appoint an impartial panel of judges to do the actual selection of winners. Post entries on a bulletin board; winners might be posted in the hallway outside your room.
3. There is no need to use graph paper. Instead, indicate that FORWARD 6 means to draw a 6 cm line in the direction the pen is pointed. All of the previous activities can be repeated using plain paper and a one cm unit of measure. If you want the pictures to come out as before, the line width must also be one cm. More detailed pictures are possible if the unit of measure is taken to be one mm.
4. In Logo the statements RIGHT and LEFT are accompanied by an angle in degrees. Thus, instead of just saying RIGHT, one says RIGHT 90 to get a 90 degree clockwise turn. This allows other angles of turn. LEFT 30 would be a counterclockwise turn of 30 degrees. All of the previous activities can be

25

repeated using plain paper and these modified RIGHT and LEFT statements. Students will need protractors to measure angles.

5. Logo is now available on quite a variety of microcomputers. Arrange for a Logo language microcomputer to be made available to your students. Let them experiment with running demonstration programs. Give each student a printout of a couple of simple Logo programs and the output produced from running these programs. If time permits, give instruction in the rudiments of writing Logo programs for the computer and allow your students to write some simple programs.





Activity #7

ACTING OUT THE COMPUTER

SUBJECT AREA: Computer Literacy.

SKILLS TAUGHT: Computer oriented vocabulary (parts of a computer system), functions of a computer system, computer processing of a program.

TIME: Forty to fifty minutes.

GROUP SIZE: Thirteen to fifteen students.

FACILITIES AND MATERIALS: Two desks, 14 chairs, blackboard or overhead projector or chart paper, appropriate writing implement, tagboard for labels, 3x5 cards for program statements and data, pointer or meter stick, paper and pencil.

OBJECTIVE: To teach students the parts of a computer system and how they function.

BACKGROUND:

As students use and study computers, they form "mental models" (mental images, pictures in their heads) of what a computer is and how it functions. These mental models help students to understand computers and provide foundations for future learning. It is important for a teacher to realize that all students form these mental models and that the models formed by different students may vary considerably. One purpose of formal instruction is to ensure that the mental models resemble reality enough to serve as a vehicle for communicating with other people and as a model for further learning.

Acting Out the Computer is designed for students who have had some hands-on experience with a microcomputer or other computer system. Students at this level probably have a mental model of a computer as being a collection of parts with some typewriter-like characteristics and some T.V.-like characteristics. Their mental models may also incorporate images related to the software used with the computer.

This activity assumes that students have done one or more activities designed to introduce the idea of a procedure. A student's mental model for a procedure may include images of the steps involved in making a peanut butter and jelly sandwich, tying a shoe or following a map. These mental models may be interwoven with those for a computer system, since the idea of a procedure was introduced in the context of learning about computers. The purpose of this activity is to provide students with another mental model of a computer system. This model is designed to give better insight into the actual internal functioning of a computer. Hence, it provides students with increased understanding of the types of procedures that computers can carry out.

A computer system has five main functional components: input unit, output unit, memory (storage) unit, arithmetic/logic unit and control unit. Several of these components often are physically combined. For example, in a microcomputer the arithmetic/logic unit and the control unit may be combined into a single chip. The chip may even contain a substantial amount of memory.

A typical microcomputer system includes the control unit, arithmetic/logic unit, primary memory and a keyboard for input in one physical unit. A T.V. set or monitor for output and a cassette recorder or disk drive for secondary storage completes the system. However, for the purposes of this activity, the five main components will be presented as separate and distinct.

28

The purpose of an input unit is to get information into a computer system. A typewriter-like keyboard is a common input device. But there are many other input units such as a joystick or hand controller, a touch panel or a sensing device such as in a measuring instrument. An appropriate mental model is one of words, numbers and measurements from scientific instruments flowing on a conveyor belt into a computer system.

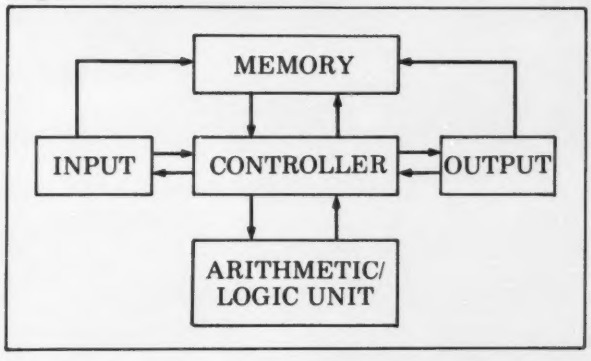
The purpose of an output unit is to get information out of a computer system. A typewriter-like printer or a T.V. display screen are common output devices. But output might be voice, music or control signals to a piece of assembly line machinery. Again, an appropriate mental model is one of words and numbers flowing on a conveyor belt out of a computer system.

Memory is storage space for the temporary or permanent storage of input and for results produced by the computer. The input consists of instructions to be carried out by a computer and data to be processed by the instructions. In many ways computer memory is like a set of mailboxes, each capable of holding a small amount of information. A mental model for memory might consist of a long row of mailboxes, each containing a piece of paper. Each mailbox is a memory location. Whatever is written on the piece of paper in a particular mailbox is the contents of the memory location represented by that mailbox.

The arithmetic/logic unit (ALU) actually carries out instructions. When given an instruction, the ALU can perform an arithmetic operation such as addition or multiplication. It can also perform logical operations such as deciding if one letter comes before another letter in the alphabet. A mental model of an ALU might include a super fast calculator with the ability to make simple comparisons between pairs of letters or numbers.

The control unit directs the operations of all the other parts of the computer. A timing device or computer clock is used to regulate this task. Another important function of the control unit is the processing of the instructions specified in a computer program. This task is performed by the instruction processor. One might have a mental model of a program written on a number of cards, one instruction per card. The cards are stored in a row of mailboxes (memory locations). Instruction execution takes two steps. In the first the control unit fetches an instruction from a mailbox for the instruction processor to analyze. In the second step the instruction is executed. Execution may involve the fetching of data from primary memory. The control unit performs the data fetch from primary memory and sends it to either the ALU to be processed or to an output device. The results of the ALU's processing are stored and may also be sent to an output device. The control unit and the arithmetic/logic unit together are referred to as the central processing unit (CPU).

Figure 1.



This activity is based on these articles from *The Computing Teacher:*

Kathleen Beyer, "Acting Out the Computer," Vol. 6, #4.

Barbara Feddern, "Computer Memory Board: A Tool for Teaching Basic Programming to Elementary Students," Vol. 8, #6.

Barbara Feddern, "Playing Computer: A Computer Literacy Activity for the Elementary Grades," Vol. 9, #2.

PROCEDURE

Introduce the components of a computer system by discussing machines or objects that accept input, process that input, and output some result.

29

Here are some examples that you might use:

Machine/Object	Input	Processing	Output
car wash	dirty car	washing	clean car
blender	milk, ice cream, chocolate syrup	blending	chocolate milkshake
tape deck	tape	playing	music
stereo	record	playing	music
calculator	numbers and operation	adding, subtracting, multiplying, dividing	answer

Make a list of the examples you present and add student examples to the list. Someone will probably mention a computer. If not, you can add that a computer is also an example of a machine that accepts input, processes it, and then outputs the result.

Next, you might want to pose the following questions about the list: Does the machine or object ever store the input for future use? In these examples, no, unless the calculator has a memory. This is different from a computer, where the input is stored in memory. Does the input ever remain unchanged? Yes, for the tape deck and stereo. This is similar to a computer, where the input from tapes and disks still exists in its original form.

Now you can narrow the discussion to the input, processing and output of the computer. On the board or overhead, list:

INPUT PROCESSING OUTPUT

Discuss the following questions and add details under the appropriate heading:

What processing does a computer do?
- accepts input and stores it—stores input
- analyzes type of input—analyzes input
- processes depending on type—performs arithmetic operations and logical operations

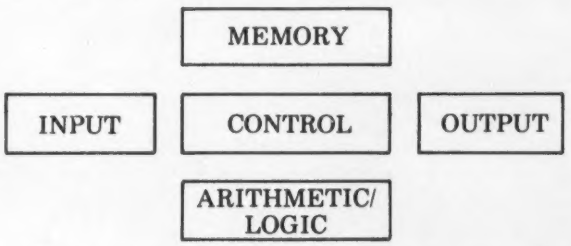
What can you input to a computer?
- letters
- numbers
- words
- sentences
- commands (RUN, SAVE, LIST)
- single keystrokes (SPACE BAR, RETURN, ESC, etc.)
- spoken words
- measurements or settings from separate equipment

What are examples of computer input devices?
- tape recorders, keyboards, disk drives, etc.

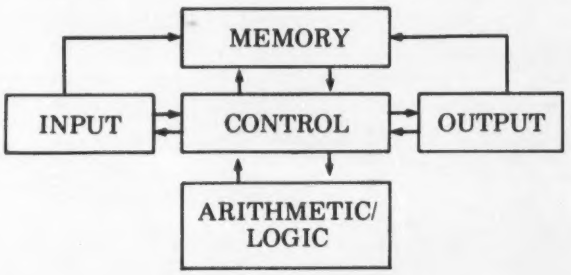
What does a computer output?
- results
- requests for input

What are examples of computer output devices?
- T.V. or monitor
- printers
- tape recorders
- disk drives

Now, you can focus on the components of a computer system that handle the processing. We have said that the computer stores the input so it must have a memory. Add MEMORY above PROCESSING on the board. Give an appropriate mental model (row of mailboxes). We have said that a computer analyzes input. That is one of the functions of the control unit. Add CONTROL in place of processing. Finally, we have said that a computer performs arithmetic and logical operations—that is the function of the arithmetic/logic unit. Add ARITHMETIC/LOGIC below CONTROL. Provide an appropriate mental model (super fast calculator).



Finally, to this diagram fill in the arrows to indicate the lines of communication between the units. (See Part III. Setting the Stage.)



30

Exactly what happens when a user enters a program into a computer? Let's act out the computer to find out.

I. Cast (13 to 15 students)

INPUT—one student
OUTPUT—one student
ARITHMETIC/LOGIC—one student
MEMORY LOCATIONS—eight to ten students
INSTRUCTION PROCESSOR—one student
USER—one student

II. Props

1. Tagboard labels for all components: INPUT, OUTPUT, ARITHMETIC/LOGIC, MEMORY and INSTRUCTION PROCESSOR and label for state of the computer—ON on one side and OFF on the other.

2. Two desks and 13 chairs.

3. Paper and pencil on one desk for the ARITHMETIC/LOGIC unit.

4. Blackboard and chalk, or overhead projector and pen, or chart paper and marker for OUTPUT.

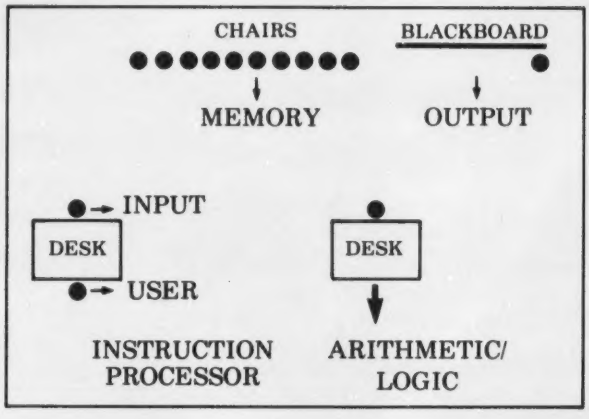
5. 3x5 cards preprinted with one program line on each card and extra cards to hold input values. Extra cards should be located on the desk for INPUT.

6. Pointer or meter stick for INSTRUCTION PROCESSOR to hold.

7. Instructions (one instruction per 3x5 card):

   #1 Print "This is a program to add two numbers."
   #2 Print "Give me two numbers."
   #3 Get two numbers from the user.
   #4 Add the first number to the second.
   #5 Print "The answer is" contents of memory location #8.
   #6 Stop

III. Setting the Stage:



INPUT: Student sitting in a chair behind a desk who will take 3x5 cards from USER. INPUT communicates with MEMORY at the direction of the INSTRUCTION PROCESSOR.

OUTPUT: Student sitting by the blackboard ready to write. OUTPUT communicates with MEMORY at the direction of the INSTRUCTION PROCESSOR.

MEMORY: Ten chairs for MEMORY LOCATIONS (8 are actually needed). MEMORY communicates with INPUT, OUTPUT, and ARITHMETIC/LOGIC at the direction of the INSTRUCTION PROCESSOR.

ARITHMETIC/LOGIC: Student sitting at a desk with paper and pencil. ARITHMETIC/LOGIC communicates with MEMORY at the direction of the INSTRUCTION PROCESSOR.

INSTRUCTION PROCESSOR: Student with pointer or meter stick. The INSTRUCTION PROCESSOR is the part of the control unit that directs the other units in the fetching and executing of instructions.

USER: Student sitting in front of the INPUT desk and communicating with INPUT via instructions and data on 3x5 cards.

IV. Script

USER: (Comes in and turns on the computer by turning the card on INPUT's desk to ON.)

INSTRUCTION PROCESSOR: (Points to the MEMORY LOCATIONS. The students file in and sit down.)

31

USER: (Enters the program by handing the 3x5 cards containing the instructions to INPUT.)

INSTRUCTION PROCESSOR: (Points to the first MEMORY LOCATION.)

INPUT: (Gives instruction #1 to the first MEMORY LOCATION.)

INSTRUCTION PROCESSOR: (Points to the second MEMORY LOCATION.)

INPUT: (Gives instruction #2 to the second MEMORY LOCATION.) (This process continues until all the instructions are stored.)

INSTRUCTION PROCESSOR: (Points to the first instruction.)

MEMORY LOCATION #1: (Stands and reads the 3x5 card.) #1 Print "This is a program to add two numbers."

OUTPUT: (Prints the message within the quotes on the board.)

INSTRUCTION PROCESSOR: (Points to the second instruction.)

MEMORY LOCATION #2: (Stands and reads the 3x5 card.) #2 Print "Give me two numbers."

OUTPUT: (Prints the message within the quotes on the board.)

INSTRUCTION PROCESSOR: (Points to the third instruction.)

MEMORY LOCATION #3: (Stands and reads the 3x5 card.) #3 Get two numbers from the user.

INPUT: (Waits for the USER to enter two numbers.)

USER: (Writes any number on one card. Writes any number on a second card. Then hands the two cards to INPUT.)

INSTRUCTION PROCESSOR: (Points to MEMORY LOCATION #7.)

INPUT: (Gives the card with the first number to MEMORY LOCATION #7.)

INSTRUCTION PROCESSOR: (Points to MEMORY LOCATION #8.)

INPUT: (Gives the card with the second number to MEMORY LOCATION #8.)

INSTRUCTION PROCESSOR: (Points to the fourth instruction.)

MEMORY LOCATION #4: (Stands and reads the 3x5 card.) #4 Add the first number to the second.

INSTRUCTION PROCESSOR: (Points to MEMORY LOCATION #7.)

MEMORY LOCATION #7: (Hands the card with the first number to ARITHMETIC/LOGIC who copies it on the paper and gives it back. MEMORY LOCATION #7 then returns to chair #7.)

INSTRUCTION PROCESSOR: (Points to MEMORY LOCATION #8.)

MEMORY LOCATION #8: (Hands the card with the second number to ARITHMETIC/LOGIC who computes the sum of the two numbers. ARITHMETIC/LOGIC then marks out the second number and puts the answer on the card and hands the card to MEMORY LOCATION #8 who returns to chair #8.)

INSTRUCTION PROCESSOR: (Points to the fifth instruction.)

MEMORY LOCATION #5: (Stands and reads the 3x5 card.)#5 Print "The answer is" contents of memory location #8.

OUTPUT: (Prints the message within the quotes on the board.)

INSTRUCTION PROCESSOR: (Points to MEMORY LOCATION #8.)

MEMORY LOCATION #8: (Stands and reads the 3x5 card.)

OUTPUT: (Prints the answer on the board.)

INSTRUCTION PROCESSOR: (Points to the sixth instruction.)

MEMORY LOCATION #6: (Stands and reads the 3x5 card.) #6 Stop

EXTENSIONS

1. Have students act out the computer for the following set of instructions:

   #1 Print "This program tells which of 2 names comes first when they are alphabetized."

   #2 Print "Give me two names."

#3 Get two names from the user.

#4 Compare the first to the second and return the name that comes first in the alphabet.*

#5 Print "The first name is" contents of memory location #8.

#6 Stop

*Note: During execution of instruction #4, when ARITHMETIC/LOGIC crosses out the name on the card from the second data location to write the name that comes first in the alphabet, s/he will do this even if the names are the same, i.e. the alphabetical order was correct when the names were entered.

2. You can modify the memory of your "computer" to accommodate larger programs. One method is to construct a large pocket chart that will hold approximately 15 instructions and data items. One student will now be in charge of memory (pocket chart). S/he will load the instructions into each slot, read the instructions that the instruction processor points to and deliver the required information to the ALU. Call this student MEMORY instead of a numbered MEMORY LOCATION. Have the students act out the computer for the following set of instructions:

#1 Print "This is a program to add two numbers."

#2 Print "The computer will work problems until you tell it to stop."

#3 Print "Give me a number."

#4 Get one number from the user.

#5 If the input is STOP, then go to instruction #11.*

#6 Print "Give me a number."

#7 Get another number from the user.

#8 Add the first number to the second.

#9 Print "The answer is" contents of memory location #13.

#10 Go to instruction #3.

#11 Stop

*Note: Script after instruction #5 is read:

INSTRUCTION PROCESSOR: (Points to MEMORY LOCATION #12.)

MEMORY: (Takes the 3x5 card from MEMORY LOCATION #12 to ARITHMETIC/LOGIC.)

ARITHMETIC/LOGIC: (Copies the contents of the 3x5 card. Looks to see if it is stop. If it is stop s/he replies, "Yes" to the INSTRUCTION PROCESSOR. Otherwise, s/he replies, "No."

INSTRUCTION PROCESSOR: (If the answer is "yes," points to MEMORY LOCATION #11. If the answer is "no," points to MEMORY LOCATION #6.)

Note: Memory locations #12 and #13 will be reused each time a new problem is entered.)

3. Have students divide up into groups of 3 or 4. Have each group write a program to add up as many numbers as the user wants for one problem. Have students act out the computer for some or all of the programs to see if they work or not.

Note or hint: Store the answer after each addition in the location of the first number entered instead of the second. Use the instruction—add the second number to the first.)

Sample:

#1 Print "This program adds as many numbers as you want."

#2 Print "The computer will add up numbers until you tell it to stop."

#3 Print "Give me a number."

#4 Get one number from the user.

#5 Print "Give me a number."

#6 Get one number from the user.

#7 If the input is to STOP, go to instruction #10.

#8 Add the *second number to the first.*

#9 Go to instruction #5.

#10 Print "The answer is" contents of memory location #12.

#11 Stop

Note: The running total is being stored in memory location #12. The new number to be added is stored in memory location #13.

4. This extension is useful if students are also learning some elementary programming. You will want to introduce the idea of a variable. A logical, concrete approach is necessary for elementary and middle school students to understand variables and their function in a computer program. Barbara Feddern in her

article "Computer Memory Board: A Tool for Teaching Basic Programming to Elementary Students" (TCT Vol. 8, #6) has suggested adding a "memory board" to your computer system. The memory board represents the section of memory (rows of mailboxes) that is used to store variables. This section of memory is in addition to the section that stores instructions. Each mailbox is given a name, but the contents may vary from time to time. A special memory board may be constructed from the directions given by Barbara Feddern in her article or a separate pocket chart may be used. Each slot should contain two cards—one for the variable name and one for the content which can change. The computer model now becomes a useful tool for debugging student programs and for illustrating constructs such as IF-THEN and FOR-NEXT.

Start with a very simple program like the following:

#1 A=24
#2 B=10
#3 C=A*B
#4 PRINT C

Briefly, the steps would be:

1. The user hands the instruction cards to INPUT.
2. INPUT hands the cards to MEMORY who places them one by one into the pocket chart at the direction of the INSTRUCTION PROCESSOR. As each instruction is placed into the pocket chart, it is examined to see if it contains a variable. If it does, then a card from the second pocket chart is labeled with the same variable name, and the second card is labeled with the contents of that variable if it has been determined.

Example:

#1 A=24
#2 B=10
#3 C=A*B
#4 Print C
#5 Stop

A	24
B	10
C	

3. The INSTRUCTION PROCESSOR points to each instruction, and MEMORY reads the instruction.
4. At instruction #3 MEMORY at the direction of the INSTRUCTION PROCESSOR would take the card containing "24" to ARITHMETIC/LOGIC to copy. Then MEMORY would return that card and take the card containing "10" to ARITHMETIC/LOGIC to copy. ARITHMETIC/LOGIC would multiply 24 and 10. MEMORY would take the blank card in slot C to ARITHMETIC/LOGIC who would copy the answer on the card. MEMORY would return the card to the pocket chart.
5. After MEMORY reads instruction #4 s/he would read the value of C to OUTPUT who prints the answer on the board.

34

Activity #8

CATEGORIES

SUBJECT AREA: Mathematics, Science, Social Studies.

SKILLS TAUGHT: Critical thinking, categorization of objects into sets with like characteristics.

TIME: Thirty to forty minutes.

GROUP SIZE: Whole class.

FACILITIES AND MATERIALS: Paper and pencil for each student.

OBJECTIVE: To introduce students to the topic that computer scientists call sorting and searching. Students will learn that there are many ways to divide a collection of objects into sets and that numerical values can be used in categorization schemes for social science-type data.

BACKGROUND:

Computers work with letters as well as digits, with words as well as numbers. But often it is convenient, or even necessary, to take non-numerical data and assign numerical values to it.

As an example, one may be doing a study involving boys and girls, left-handed and right-handed students, fifth graders and sixth graders. One might arbitrarily use the coding scheme:

Sex
1=Male
2=Female

Handedness
1=Left-handed
2=Right-handed

Grade in school
1=Fifth grade
2=Sixth grade

Under this coding scheme a person categorized as 2,1,2 would be a left-handed girl in the sixth grade. A person categorized as 1,2,1 would be a right-handed fifth grade boy.

It is important to realize that the coding scheme is completely arbitrary. The fact that a female is a 2 rather than a 1 in no sense implies that the female is twice as good as the male or second to the male. The coding scheme is strictly for convenience in dealing with the data, and such coding schemes are particularly useful when working with a computer.

More generally, categorization is fundamental to much of science and social science. For example, the voters in a state might be divided into categories based on age, income, education, political party and sex. This can be done by assigning numerical codes to data gathered in a questionnaire and using a computer to sort people into categories. An analysis of voting characteristics and interests of some of the larger groups might help a politician formulate positions on various issues.

PROCEDURE:

Have each student write the numbers 1 through 6 down the left side of a sheet of paper. Each student is to fill in answers to six questions using the data coding scheme described

below.

1. Age in years at last birthday. Write down a 1 if your age is an odd number and write down a 2 if your age is an even number. (Students are not supposed to write down their age. Rather, they are to take the piece of data that is their age and classify it as odd or even. Then record a 1 for odd or a 2 for even.)
2. Sex. Write down a 1 for female or a 2 for male.
3. Handedness. Write down a 1 for left-handed or a 2 for right-handed. Note that this coding scheme does not allow for the possibility that a student is truly ambidextrous.
4. Height to nearest centimeter. (Alternatively, have everyone use height to nearest inch.) Write down a 1 if the height is an odd number or a 2 if the height is an even number.
5. Estimate weight to nearest kilogram. (Alternatively, have everyone use weight to nearest pound.) Write down a 1 if the weight is an odd number or a 2 if it is an even number.
6. Total number of brothers/stepbrothers and sisters/stepsisters. In this case record the total number, rather than coding it as odd or even.

Next, help your students get used to reading numerical values and responding to categorization questions. Ask questions such as the following:

- Will all students who responded 1 on question number 2 please stand up? (You expect all girls in the class to stand, and all boys to remain seated.) After correcting errors, have all those who are standing be seated.
- Will all students who responded with a 3 or larger number on question 6 raise both hands? (If you define "large family" to mean a family with four or more children, then every student holding up both hands is from a large family.)
- Will all students who responded 2 on both questions 2 and 3 please raise their left hand? (You expect all right-handed boys to raise their left hand.)

Continue with the above activity for about five minutes. You want all students to become comfortable with the process of responding to questions such as those illustrated.

Next, have each student find the sum of the six answers they recorded on their sheets. Call this sum a student's score. Ask questions such as the following:

- Who has the lowest score? (More than one student may have the lowest score.) Are there games in which the person or team with the lowest score wins? Have the class try to name some. What happens if more than one person or team gets the lowest score?
- Who has the highest score? (Again, more than one student may have the highest score.) List games in which the person or team with the highest score wins. This could lead to a discussion of why most games are designed so the person or team with the highest score wins.
- Does any student in the class have the exact score of 5? Such students must be left-handed females with no siblings. How do we know this? What else do we know about these students?
- Does any student have a score of exactly 6? If a boy has a score of 6, how many siblings does he have? If a right-handed student has a score of 6, is the student a boy or a girl?

EXTENSIONS:

1. If the students know one or more meanings of the word "average"—that is, mean, median or mode—the activity can be continued by computing the class mean, median or mode. The most common meaning of the word average is mean. This is the sum of all the scores divided by the number of scores.

   It is unlikely that any student will have a score that is exactly the class mean, but you can determine whose score is closest to the mean. This can lead to a discussion of the meaning of a statement like "this is an average student." The meaning is clearly dependent on the coding and classification scheme being used.
2. There are many other criteria that can be used in this type of activity. For example, one might assign numerical values to hair color, with a 1 for red hair, 2 for blond or light-colored hair, and a 3 for brunette or dark-colored hair. By adding several new criteria and/or using them to replace initial criteria, you can repeat the entire activity.

36

3. The assignment of numerical values to the data was quite arbitrary. We might just as well have coded a male as 1 and a female as 2; we might have coded a right-handed student as 1 and a left-handed student as 2. Make several such changes in the coding scheme, and have students note how their recorded values change and how the answers to the questions you have asked change. If you have carried out Extension 1, have students note how their "scores" are changed by a change in the coding scheme. This helps to make it clear that having a particular score has nothing to do with being good or bad.

4. Suppose that two students have exactly the same set of numerical values. They would be the same sex, the same handedness, and have the same number of siblings. But they would not necessarily be the same age, height or weight.

Suppose we want to find people in the school who are "nearly alike." We might want to use criteria such as sex, handedness, age rounded to the nearest year, grade, hair color, number of siblings, and so on. Computerized dating services use criteria such as these. Set your class to work on deciding on a meaning of "nearly alike." After coming up with suitable criteria, they may wish to categorize all students in the school to find the pair that is most nearly alike.

Activity #9

COMPUTER LITERACY REFERENCE SEARCH

SUBJECT AREA: Language Arts, Social Studies, Library Skills.

SKILLS TAUGHT: Library research skills, use of an encyclopedia index, report writing.

TIME: Two to three class periods to carry out the activity and additional time for students to prepare written or oral reports.

GROUP SIZE: Whole class or small groups.

FACILITIES AND MATERIALS: The school library/media center or classroom, at least one comprehensive set of encyclopedias and a supply of 3x5 index cards. Use the help and expertise of the school media specialist!

OBJECTIVE: To learn that encyclopedias are an excellent source of information about the history and applications of computers. To develop skill in using a well-indexed encyclopedia. To practice report writing skills such as note taking, outlining and actual writing.

BACKGROUND:

Although computers are a relatively recent tool in elementary and middle school education, the topic of computers has a long history, beginning with the abacus nearly 5,000 years ago. This history can be traced through the development of calculators in the 1600s, attempts to build a computer in the 1800s, the first electronic digital computers developed during the 1940s and then into more recent times. Computers have been mass produced since 1951 and have been commonly available in business, government, research and higher education for about 20 years.

You may be surprised at the amount of information about computers and computer applications that can be found in a recent set of encyclopedias. The purpose of this activity is to introduce you and your students to this excellent source of information about computers.

PROCEDURE:

For each encyclopedia set you will be using, photocopy the index page containing "Computer" and the list of "See Also" or "Related Topics" at the end of the entry. Duplicate a copy for each student in the class.

If your students have not received instruction in using an encyclopedia, you or the media specialist can use the topic "Computers and their Applications" to give some instruction. If your students have had such instruction, only a brief review may be needed.

Your students will probably know that they can find something about computers under the heading Computer in the C volume of the encyclopedia. But where else might they find information?

Challenge your students to think about what they might find on computers and their applications in other volumes of the encyclopedia. Artificial Intelligence, for instance, has to do with developing computers that can carry out tasks that appear to require intelligence. Automation is more and more dependent upon computers, and robots are computerized machines often used in automation.

When they have developed a list, either indi-

vidually or in small groups, have them check the "See Also" references for additional topics.

Now, using the list of "See Also" references, have each student select one or two appropriate volumes from the sets of encyclopedias and prepare an index card listing for computer-related references in those volumes. Each reference should be in a standard bibliographic format (be sure to show students an example), and each should contain a sentence or two summarizing the information available.

Once students have completed the above activity, each student can select a specific computer-related topic, research it (preparing index cards as appropriate) and prepare a written or oral report.

EXTENSIONS:

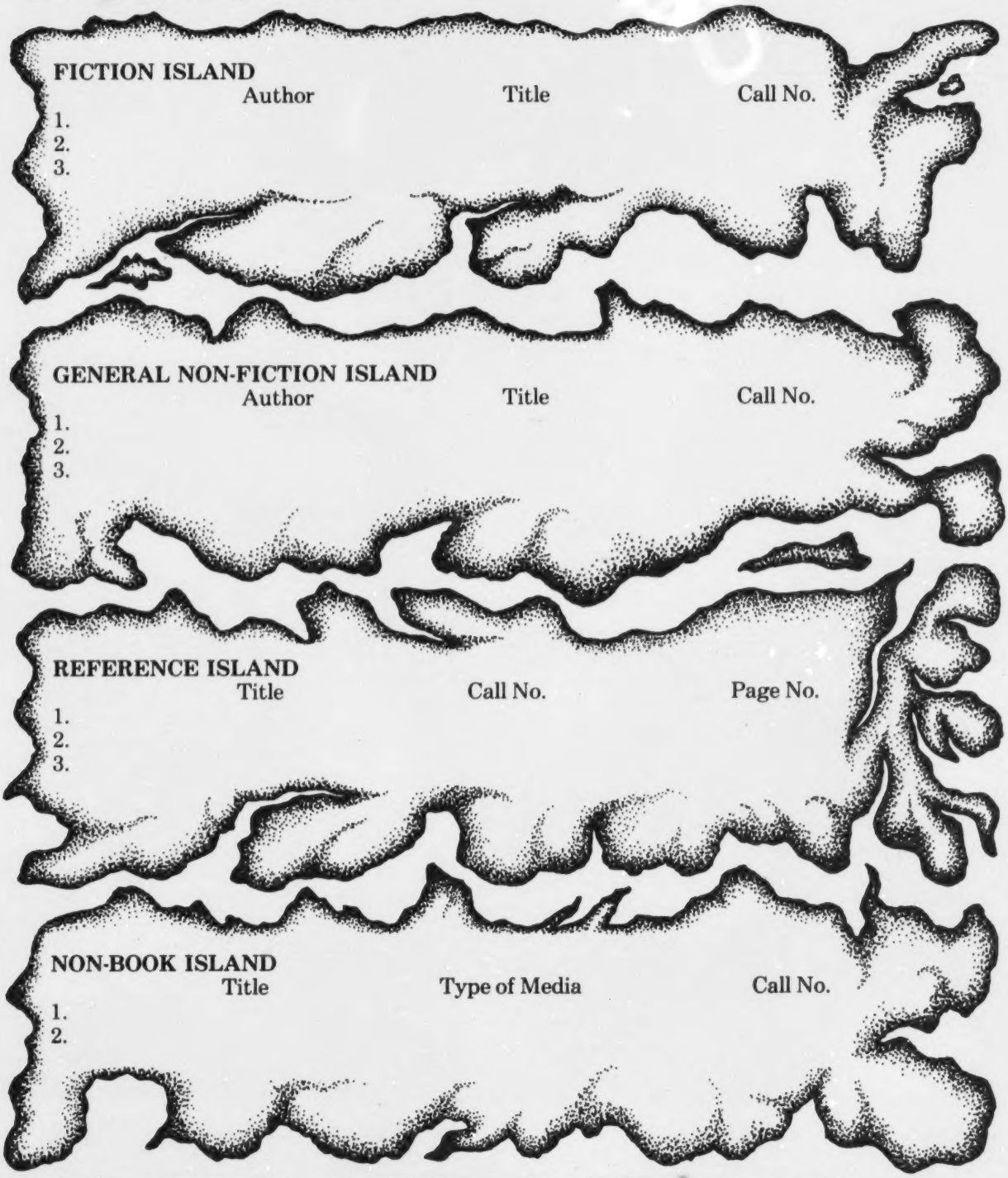
1. Find encyclopedias produced by several different publishers. Divide the class into teams, one for each set of encyclopedias. Have each team develop as extensive a set of index cards for computer-related references as possible. Were they able to find appropriate topics which were not included in the "See Also" list? Use this activity to help students compare encyclopedias. Do all sets seem to contain the same entries? Are the entries of the same approximate length and complexity? Are the indexes equally complete and useful?
2. Locate an encyclopedia set from the 1970s, 1960s, 1950s, 1940s, 1930s, etc. Have each student select a computer topic such as the one used in their written or oral report in the main part of this activity. The students are to research their topics in older and older encyclopedias to see the types of changes that have occurred in the references over the years. For most topics, students will find that if they go back far enough they will be unable to find an entry.
3. Have a "Library/Media Center Treasure Hunt" to find computer references in as many different types of books as possible. Students can look for references in fiction books; general non-fiction books; reference books; and non-book sources of computer information such as filmstrips, slides, videotapes or kits.

   There are many ways to conduct this treasure hunt. You may first want to do such a search yourself, using the card catalog and the media specialist for assistance. A master to photocopy is included after the Extensions, or you may want to prepare your own. (You can even burn the edges, roll them up and tie them for an added "treasure map" effect!) Small groups of students, working individually or in pairs, can fill in their maps using the card catalog and their knowledge of specific books and other information on computers. When they turn in their completed maps, give them a separate slip of paper which will lead them individually to a treasure (sugarless gum, small bag of popcorn, stickers, etc.). Example: "Look behind the book, *Little Women.*"
4. Have students compile a list of the findings on the materials about computers in their school library/media center into a Computer Bibliography, to be left in the library as a reference guide for other students. Discuss the kinds of information about computers that were discovered. Were students able to locate any computer information without consulting the card catalog? Did the subject headings seem adequate? Do they have any recommendations for purchase?
5. Explore the concept of computerized card catalogs. If a high school, college or university library in your area uses such a system, try to arrange a visit and demonstration. Explore this topic in terms of ease of use, cost, updating, immediate and/or potential problems. A few students might want to research this topic further and present their findings to the school media specialist.

Name _____
Date _____
Class _____

COMPUTER INFO TREASURE MAP

Find the information on computers located on each island. Use the card catalog and your own knowledge about computer resources.



FICTION ISLAND

	Author	Title	Call No.
1.			
2.			
3.			

GENERAL NON-FICTION ISLAND

	Author	Title	Call No.
1.			
2.			
3.			

REFERENCE ISLAND

	Title	Call No.	Page No.
1.			
2.			
3.			

NON-BOOK ISLAND

	Title	Type of Media	Call No.
1.			
2.			

Trade in your completed Computer Info Treasure Map for directions to a surprise.

Activity #10

CARD-SECTION GRAPHICS

SUBJECT AREA: Mathematics, Computer Graphics.

SKILLS TAUGHT: Developing and following directions; Cartesian coordinates.

TIME: Sixty to ninety minutes.

GROUP SIZE: Whole class, or a larger group.

FACILITIES AND MATERIALS: Each student will need a large poster-board card, like those used in the card section at many college sporting events. All cards should be the same size and colors, with contrasting colors on the two sides. A video camera is a fun addition but is not necessary.

OBJECTIVE: Students will develop and display card-section pictures; in the process of doing so they will learn about computer graphics and increase their knowledge about Cartesian coordinates.

BACKGROUND:

Computers can be used to create, manipulate and display pictures. This is called computer graphics. If a microcomputer is being used, the display mechanism will likely be a color or black and white television set.

In either case, it is possible to program the computer to output pictures and graphs. The graphics may be made of small rectangles or they may be made of very small dots. The use of small rectangles is called low resolution graphics and the display screen may be divided into a 40 row by 40 column array of small rectangles or squares. The use of very small dots is called high resolution graphics and the display screen may be divided into a 200 row by 300 column array of dots. (In both cases the size of the arrays varies with the particular brand and model of microcomputer being used.)

While one can output more intricate and detailed graphics on a high resolution system, the general ideas of low resolution graphics and high resolution graphics are the same. One can think of the display screen as being divided into a grid of small rectangles. Under computer control one can set the color (or, black-white) of each grid square. The programmer decides upon the pattern of colors and squares, and then writes a program to cause the computer to actually carry out the task.

In a very simple computer graphics programming language, the programmer will use instructions such as the following:

```
COLOR = 5
SET 34,18
SET 35,18
SET 35,19
```

In this short example the color is set to color number 5. (Each of the available colors has a color number.) The next three instructions direct the computer to output three small rectangles or dots. The first is to be in the grid location with Cartesian coordinates (34,18); the second is in the grid location with coordinates (35,18); the third in the grid location with coordinates (35,19).

A more sophisticated graphics programming language may have features such as illustrated above and also the ability to easily specify a complete line segment. For example,

```
DRAW 20,12 TO 35,18
```

would instruct the computer to draw a line segment from the grid location (20,12) to the grid

location (35,18). The computer would calculate what grid locations this line segment actually passes through and produce the desired display.

Mathematicians generally use a pair of numbers to designate a Cartesian coordinate location. With young students you will likely want to use a combination of letters and numbers. Two possible schemes are illustrated in the grids given below. Each is a seven row by six column grid.

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

	1	2	3	4	5	6
A						
B						
C						
D						
E						
F						
G						

When using Cartesian coordinates, mathematicians generally have the first coordinate specify distance along the horizontal axis (that corresponds to a column number), and the second coordinate specify distance along the vertical axis (that corresponds to a row number). Following this idea, a point in the left grid would be specified by naming a letter (distance along horizontal axis-column specification) followed by a number (distance along vertical axis-row specification). It is important that you be consistent, since eventually students will learn to use a number pair to identify a point in Cartesian coordinates.

PROCEDURE:

Introduce your students to the idea of a card-section at a sporting event. You can probably find a movie showing a card section in action or some photographs. A nearby college or television station may be able to help you find appropriate pictures. (The college may have the posterboard cards mentioned in the materials section.)

Draw a grid on the blackboard or on an overhead projector. You can decide upon an appropriate grid size; likely you will want the number of grid locations to be the same as or slightly less than the number of students in your class, and you will want the grid to be nearly square.

Letters and pictures can be made by coloring in part of the grid. On a six row by five column grid the letters J and M could be displayed as follows.

```
. . . . #        # . . . #
. . . . #        # # . # #
. . . . #        # . # . #
. . . . #        # . . . #
# . . . #        # . . . #
. # # # .        # . . . #
```

Here we have used dots to indicate blank grid locations and the # symbol to indicate a colored grid location. In a card section the blanks would be one color such as white and the #'s would be a contrasting color such as red.

Work with your class to help them learn how to select an appropriate grid size. For example, suppose there were 24 students in your class. The number could be arranged in a 1 x 24 grid, a 2 x 12 grid, a 3 x 8 grid, a 4 x 6 grid, a 6 x 4 grid and so on. Which arrangement is best for forming letters? Note that this part of the activity is a useful application of factoring.

The number of students in your class probably will not exactly fit a good grid size. You may need to borrow a few students from another class or use the extra students as leaders and aides. Perhaps you will want to have your leaders decide upon the grid notation system to be used (for example, letters along the side and digits along the top). Have your aides help you to prepare ditto master sheets containing grids. Each student will need a sheet containing a number of grids.

Assign each student two or three letters to draw out on their grids. Distribute the whole alphabet among the students so that each letter gets worked out by a couple of students. This may lead to some small group discussions on the "best" way to make a particular letter. If so, you have extended this activity to include art.

Work with the entire class or the small group of leaders to decide upon a short message to be spelled out by the card section. An example might be THE FIFTH GRADE SAYS HELLO. Count the number of letters and blank spaces in the message (in this case there are 25 letters and blanks). Have each student who will be in the card section write down the integers from one up to the number of letters on a sheet of paper.

Next arrange the card section students into

the card section grid. You may want to seat the students in a set of bleachers or on some steps. Each student is to write the coordinates of their location on their sheet of paper.

The next step is to develop the card section program. Instruct the card section members to hold their cards so that they are all looking at the same colored side (say the red side, if the sides are red and white). If they are all looking at red sides, then people standing in front of the students will see only white sides.

Now "program" the first letter. To do this, read off the coordinates of the grid locations that make up the first letter. Each student whose coordinate is named is to flip his/her card and is to write the newly visible color (in our example, white) on the first line of his/her sheet of paper. All students who are not instructed to flip their cards write the color they see (that is, red) on the first lines of their sheets.

Each letter of the message is programmed in the same way. The students start a letter by all looking at the same color. The programmer tells them how to form the letter and this information gets written down on the sheets. The blank space between words also needs to be programmed.

After the entire message has been programmed, you will want to have the card section practice presenting the message. It takes quite a bit of practice to get smooth and accurate movement from one letter to the next. The ultimate test is to have the card section perform at a school assembly or sporting event, or to videotape their performance.

EXTENSIONS:

1. Have each student decide upon a message and work out the programming for the message. Thus, if a six row by four column grid is being used, each student will need to prepare 24 instruction sheets, one for each student in the card section.
2. An alternative approach to programming as described in the main activity is to provide each card section member with a program for each letter and for a blank. The student in card section B,4 has directions on what color to display for an A, what color to display for a B, what color to display for a C and so on. The student in location B,5 would have a different set of directions telling what color to display for each letter. The leader then calls out a letter of the message, gives each card section member time to read what color they are to display, and then gives the instruction DISPLAY. With some practice the display section can learn to rapidly and accurately spell out entirely new messages.
3. If you have a student who is an accomplished computer programmer, assign the student the task of writing a computer program for Extension 1.
4. If one has a very large card section then it is possible to display intricate pictures. A 20 row by 20 column piece of graph paper can simulate a 20 by 20 card section. Have students develop pictures that can be displayed by a large card section. If you like, you can assume that each card section member has several cards of different colors. This allows the displays to be multi-colored.

Activity #11

SORTING

SUBJECT AREA: Language Arts, Computer Programming, Social Studies.

SKILLS TAUGHT: Following directions; cooperation.

TIME: About thirty to forty minutes.

GROUP SIZE: About six to ten students.

FACILITIES AND MATERIALS: A fairly large open floor space. One card, about a foot square, for each student participating. Number the cards 1, 2, . . . up to the number of students; use a felt tip marker so that the numbers can be read from across the room.

OBJECTIVE: Students will understand a method for sorting data; this method can be programmed for use on a computer.

BACKGROUND:

Many applications of computers involve the storage and use of very large sets of data. A school district maintains a file containing information about every student. The file contains a student's name and address, parents' names and address, grade in school, academic record since starting school, scores on various standardized tests, medical records and so on. The school district employees' file contains information about every teacher, including payroll information, years worked for the district, present school and teaching assignment, and so on.

A typical use of such computerized data banks is to arrange some data in a specified order and then print out a report. The superintendent might want a list of teachers with ten or more years of service, ordered from greatest to least number of years of service. The Honor Society might want a list of all seniors with a grade point average above 3.5, ordered from highest to lowest. Students may want to put out a student directory, requiring a list of all students arranged alphabetically.

Sorting a list, either numerically or alphabetically, is a standard use of computers. Most people know how to do such sorting activities by hand. But often they have not thought carefully about the details involved. Also, they might not have examined and compared different sorting methods. A student majoring in computer science at a university might take a whole course on sorting methods and comparison of these methods. Which is fastest, which is easiest to program, which uses the least amount of computer memory, which works best if the data have been partially presorted? How long will it take to sort a given set of data? These are often hard questions to answer.

The social issues of large databanks containing information are very important. On the school records, for example, what information is stored, how is its correctness verified, who has access to the data and for what purposes? Suppose that a program to determine teachers' take-home pay contains an error. Teachers get underpaid or overpaid. While the error might be discovered at a later date, either situation can have serious consequences.

The activity given below is adapted from the article "Teaching First Graders How a Computer Can Sort," written by William D. Hedges and published in Vol. 8 No. 1 of *The Computing Teacher.*

PROCEDURE:

Provide students with some of the background information discussed above. The discussion might focus on school records and whether the records in your district are computerized. You may be able to get some computer printouts, showing data sorted into numerical or alphabetical order. The class discussion might lead to questions about what information should be stored in databanks, who should have access, how the data should be used and so on.

Shuffle the numbered cards and hand out one card per student. Then have the students line up in order, from lowest to highest card number. You should expect some confusion to occur in this process, but eventually, they will get themselves arranged. This provides evidence that the students understand numerical sorting.

Ask for student comments about the sorting process. What led to confusion? What sorting method did they use? Did one student emerge as the "leader," directing other students to their correct locations? Suppose that each student knew his/her number but that the sorting process had to be done while blindfolded. How could this be accomplished?

Next, have students place their numbered cards on the floor in front of their feet. Explain to the students that the next activity is to sort themselves in order according to height, from shortest to tallest. But they are to do this following precise instructions that you will give.

To begin, the student in position number 1 (that is, the student standing behind the #1 card) asks the student in position number 2, "Are you taller than me?" If the answer is no, the two students exchange positions. If the answer is yes, they do not exchange positions.

To continue, the student in position number 1 now asks the student in position number 3, "Are you taller than me?" If the answer is no, they exchange positions. If yes, they do not change positions. This process continues until the student or students in the number 1 position have questioned the students in every other position. At the end of this first "pass" through the line of students the shortest student will be in position number 1.

Now the whole process is repeated, but the student in position number 2 does the question asking, and the question is asked of the students in positions 3, 4, and so on. The idea is that the shortest student in positions 2, 3, etc. will be moved to position 2. At the end of this second pass through the line of students, the shortest student will be in position number 1 and the next to the shortest student will be in position number 2.

Now you see the pattern. In the third stage, the student(s) in position number 3 do the question asking, and the question gets asked of students in positions 4, 5, and so on. This sorting process can be used with any number of students. The reason for restricting it to 6-10 students is that this sorting algorithm is quite time consuming.

It is possible to do a mathematical analysis of sorting algorithms. The amount of time the algorithm we are using takes to sort a list varies with the square of the length of the list. For example, suppose that it takes about five minutes for eight students to be sorted. If we used the same procedure to sort 16 students (that is, twice as many students) it would take about 20 minutes (twice as many students takes four times as long). To sort 24 students (three times the original number) would take about 45 minutes (nine times the original time).

After completing the sorting process, make it clear to students that they have been following a detailed step-by-step set of directions, an algorithm, and that a computer can do the same thing. But a computer can do it many thousands of times as fast as the students.

Once students master the basic sorting algorithm, they can use it to sort themselves by weight, age or any other numeric characteristic. It is important for students to realize that the same algorithm can be used to solve all of these problems.

EXTENSIONS:

1. The same procedure can be used for alphabetical sorting. Instead of using the question, "Are you taller than me?", the question to ask is, "Does your name come after mine alphabetically?" A computer program that sorts alphabetically is very similar to a computer program that sorts numerically.
2. There are many sorting algorithms. Use the same physical setup as for the main activity. Begin by having the two students in positions 1 and 2 compare their heights, and arrange themselves in shorter to taller position (shorter in position 1). Then repeat the process, but this time for the two students in

positions 2 and 3. The shorter of the two students is to move to position 2. Continue this "comparison of pairs" process until the students in the next to last and last positions have compared themselves. The result of this first pass through the students is that the tallest student will be in the highest numbered position.

In the second pass through the line of students one begins with the students in positions 1 and 2, proceeding as above. But this pass ends after the students in the next to the next to the last and in the next to the last positions are compared. The second pass through the line of students results in the next to the tallest student being in the next to the last position.

The overall process is continued in a systematic manner until the students are sorted.

3. The activity can be extended to a whole class activity by having students sort numbered cards or cards containing words. For example, have each student prepare a set of eight 3x5 cards with a different word or person's name on each card. Each student will also need small slips of paper to number the positions 1, 2, 3, . . . 8. Now each student can carry out the alphabetical sorting algorithm (extension 1) by hand.

4. Make a deck of 3x5 cards containing the names of every student in the class, one student per card. Thoroughly shuffle the cards and then, time students as they alphabetize the deck. Be sure to thoroughly shuffle the deck between student's timing runs. If you like, make this activity into a contest, seeing who is fastest at sorting the deck. For those who are fast, are there common characteristics in their sorting algorithms? For those who are slow, are there common characteristics in their algorithms? Can a slow person become faster by learning to use a better algorithm?

5. With a setup as in Extension 3, have the students sort their cards using whatever algorithm they like. Then each student is to write down in a detailed step-by-step set of directions how to sort cards. The algorithm is to work for any number of cards, no matter what their initial order.

6. If a computer is available you might want to demonstrate the computer sorting of data. The following BASIC program will run on most microcomputers. It will handle a list of up to 50 pieces of data and sorts the data from smallest to largest using the algorithm given in the main activity.

```
100 REM THIS PROGRAM DEMONSTRATES
    NUMERICAL SORTING.
110 DIM A(50)
115 REM THE ARRAY A STORES THE DATA.
120 PRINT "HOW MANY PIECES OF DATA
    ARE TO BE SORTED?"
125 INPUT N
130 PRINT "PLEASE INPUT ONE PIECE OF
    NUMERICAL DATA AFTER EACH QUES-
    TION MARK."
135 FOR I = 1 TO N
140 INPUT A(I)
145 NEXT I
150 REM BEGIN OUTER LOOP TO SORT THE
    LIST OF DATA.
155 FOR I = 1 TO N - 1
160 REM BEGIN INNER LOOP TO SORT THE
    LIST OF DATA.
165 FOR J = I + 1 TO N
170 REM NO CHANGE IF THEY ARE IN THE
    CORRECT ORDER.
175 IF A(I)    = A(J) THEN 200
180 REM SWITCH ORDER IN THIS CASE.
185 T = A(J)
190 A(J) = A(I)
195 A(I) = T
200 NEXT J
205 NEXT I
210 REM THE LIST IS SORTED. NOW OUTPUT
    IT.
215 PRINT "THE SORTED LIST OF NUMBERS
    IS AS FOLLOWS:"
220 FOR I = 1 TO N
225 PRINT I, A(I)
230 NEXT I
235 END
```

The program can be modified to handle aphabetical data. For most microcomputer versions of BASIC, this merely requires that one replace each occurrence of A with A$. For example, line 110 would become

```
110 DIM A$(50)
```

while line 140 would become

```
140 INPUT A$(I)
```


46

Activity #12

GROWING A BINARY TREE

SUBJECT AREA: Language Arts, Mathematics.

SKILLS TAUGHT: Following directions, left/right discrimination, alphabetical and numerical sorting, computer programming.

TIME: Forty to fifty minutes.

GROUP SIZE: Seven or more.

FACILITIES AND MATERIALS: A large open floor space.

OBJECTIVE: Students will create and traverse a tree structure to sort a list into alphabetical or numerical order.

BACKGROUND:

If you haven't read the Background section for Activity 11, do so now. Activities 11 and 12 cover the same computer and information science ideas but present different algorithms to do the actual sorting of data.

You and your students are probably familiar and comfortable with data arranged in a row, column or two dimensional table. For example, you probably use a two dimensional table to record students' scores on tests and homework. The rows of the table are labeled with student names, while the columns are labeled with dates, assignment numbers or test numbers. The actual table entries are numerical scores, letter grades or check marks.

Some data do not lend themselves to such simple data structures. For example, consider your family tree. You might put yourself at the top, with two branches down to your two parents. Two branches lead down from each of your parents to your grandparents, and so on. The family tree is indeed a tree-like data structure. When "grown" from the top of the page down, it may look like the root structure of a tree. If grown from the bottom of the page up (start with your name at the bottom, draw branches to your parents at the next higher level), it may look like the limbs of a tree. A family tree structure becomes very complex if it shows brothers and sisters, aunts and uncles, nieces and nephews, and so on. Such a data structure is not conveniently represented as a two dimensional table.

Computer scientists frequently make use of tree-like data structures. They develop computer programs to examine the "nodes" (data items) in the tree to solve various problems. For example, a computerized family tree, stored as a tree-like data structure, could be used to respond to requests such as, "What was the middle name of my grandfather on my mother's side?" and "In what year did my great grandmother on my mother's mother's side die?"

A tree-like data structure can also be used to sort data alphabetically or numerically. This is illustrated by the activities below.

PROCEDURE:

Begin by discussing ideas of arranging data in rows, columns, two dimensional arrays and tree-like structures. You might have each student make a family tree showing themselves, parents and grandparents. But use care here. Many children are adopted, come from single

parent families or for other reasons cannot complete such a tree. An alternative is to illustrate using your own family tree or the family tree of some historical figure.

You will also want to review ideas of arranging data in a systematic form such as in alphabetical or numerical order. One does this to help solve some problem associated with the data. Imagine, for example, the difficulty of looking up a person's phone number if the names in a phone book were not in alphabetical order. (They instead might be arranged in increasing order of phone number or in chronological order by the date the people became phone company customers.)

Select seven students to illustrate sorting by use of a binary tree. The seven students will be sorted alphabetically by first names. In this initial instruction you will want to avoid the added confusion of having two people with the same first name.

Begin the tree-growing process by having the seven students stand at the front of the room facing the class. We will illustrate the process using the following seven names. Notice that we begin with the seven students standing in a line in "random" order.

Martha Frank Rhonda Carlos Paula Henry Tristan

Explain to the students that eventually each will be sitting down; those that are seated are "nodes" of the tree. Before becoming a node, a student must be moved to a correct location. This movement will involve "wandering" from node to node, so students through the tree are called wanderers.

Select one student to be the first node. Move this student out in front of the other students, closer to the rest of the class, and have the student be seated. In our example we have selected Martha as the first node in the tree.

Frank Rhonda Carlos Paula Henry Tristan

Martha

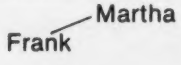
Initial situation.
One node.

Each of the remaining students will be a wanderer, one student at a time. Frank will be the first wanderer, then Rhonda, then Carlos, and so on. A wanderer goes to the first node (in this case, Martha). The wanderer and the node compare the alphabetical order of their first names. If the wanderer's name comes earlier in alphabetical order, the wanderer will move to the node's right and closer to the class. Otherwise, the wanderer will move to the node's left and closer to the class. If the move right or left brings the wanderer to another node, then the comparing process begins again. When there are no more nodes to compare, the wanderer sits down and becomes a node.

Note: There are only two directions a wanderer can move at a node—right or left. The data structure is called a binary tree because there are at most two branches from each node.

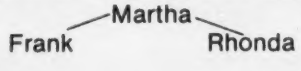
Study the example. While it may seem complex at first, it is actually quite simple. With a little instruction and practice your students will master it.

Rhonda Carlos Paula Henry Tristan



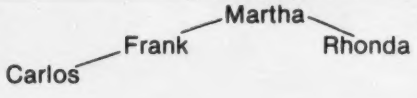
After first wanderer.
Two nodes.

Carlos Paula Henry Tristan



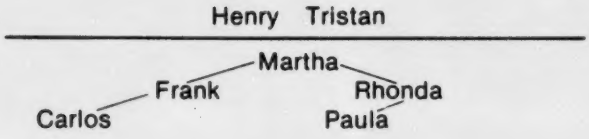
After second wanderer.
Three nodes.

Paula Henry Tristan

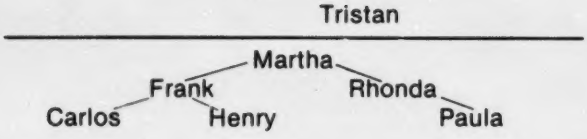


After third wanderer.
Four nodes.

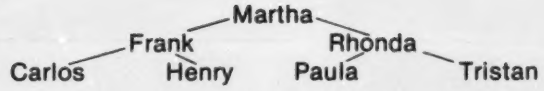
48



After fourth wanderer.
Five nodes.



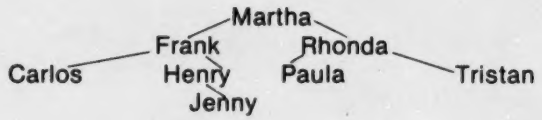
After fifth wanderer.
Six nodes.



After sixth wanderer.
Seven nodes.

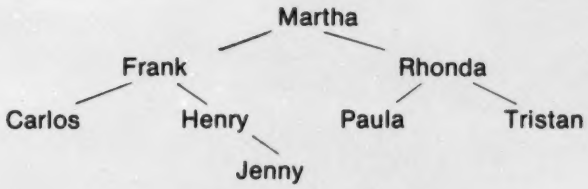
The process ends when every wanderer has become a node. To "see" the alphabetical order, think of collapsing the tree towards the class. In other words think of Frank, Martha and Rhonda walking straight forward toward the class until they are in line with the other four students. For larger and/or more complex trees it may not be so obvious that the process sorts students alphabetically. When a computer is used to create a tree-like data structure, it is also used to search the data structure. This searching process is quite similar to the initial process of creating the data structure, but proceeds more rapidly because the data structure has already been created.

In our particular example the tree "grew" evenly, with exactly two second-level nodes and exactly four third-level nodes. Obviously, this was an unusual occurrence. If an eighth student, say Jenny, now enters the tree, the result will be as follows.



After seventh wanderer.
Eight nodes.

However, since Jenny is in front of Martha, the nodes will have to spread out starting with Frank and Rhonda. Additional wanderers can be sent into the tree. The tree will continue to grow, and the wanderers will be correctly sorted into alphabetical order.



After seventh wanderer.
Eight nodes.
(Nodes had to spread out.)

The issue of two people having exactly the same name is easily handled. One just makes an arbitrary rule that if a wanderer has exactly the same name as a node, the wanderer goes to the right.

It is instructive to repeat the entire initial example with the students lined up in a different order. The resulting tree is apt to have a different shape, even though the final alphabetical ordering will be the same. After your students get the hang of it, divide your class up into groups of six to eight students and let each group try it. The original group can be helpers.

EXTENSIONS:

1. Sort each group on some other characteristic, such as height, alphabetically by last name or weight. The idea is to make it clear to students that this binary tree sorting procedure works in a variety of situations.
2. Binary tree sorting can be done as a seatwork activity. Each student makes up a deck of cards with one person's name on each card. The student shuffles the cards, selects one as the first node, and then selects the next card as the first wanderer. Subject to limitations of desk size, quite a large set of names can be sorted in this manner.

In this activity the teacher can see at a glance if a student understands the process. Alternatively, students can be asked to draw their completed trees on paper to hand in. The same set of data cards can be sorted several times, reshuffling between sorts. While the alphabetical orderings should come out the same each time, the shapes of the trees will vary considerably. For example, you might ask your students what the tree will look like if the cards happen to be exactly in alphabetical order to begin with, or exactly in reverse alphabetical order to begin with.

GLOSSARY

ALGORITHM

A finite, step-by-step set of directions guaranteed to solve a specified type of problem. Algorithms are an important part of mathematics. Students learn algorithms for the addition, subtraction, multiplication and division of whole numbers, decimal fractions and fractions. The term algorithm can also be used to describe a procedure for looking up a word in a dictionary or for alphabetizing a list of words. See also: *Procedure.*

ARITHMETIC/LOGIC UNIT

The component of a computer system that performs arithmetic operations and logical operations. See also: *Central Processing Unit (CPU).*

ARTIFICIAL INTELLIGENCE (AI)

The branch of computer science that studies how smart a machine is or can be. Computers can play chess, checkers or backgammon at the level of a state champion. They can assist in medical diagnosis, aid in foreign language translation, and even carry on a limited conversation. Research in AI has led to computers that can accept spoken commands and can "see" using a television camera for an eye. The range of problems that AI can help solve is continually increasing and overlaps greatly with the types of problems students study in school. This raises the issue of what people should learn to do mentally, what they should learn to do using pencil and paper or other simple aids, and what they should learn to do making use of a computer. This is one of the most challenging curriculum problems our schools have ever faced.

BASIC

A computer programming language developed by Kemeny and Kurtz at Dartmouth College in the early 1960s and now available for use on almost all computers. The language was specifically designed to fit the needs of college students, but the language is also widely used in business, in precollege education and by people who have home computers. (See *Logo* and *Pascal* for a brief discussion of other programming languages suited to the needs of precollege students.)

BINARY DIGIT

One of the symbols 0 or 1; often called a *bit.* The binary number system uses just two symbols to represent numbers. Starting at zero, binary counting goes 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, and so on. These correspond to the decimal notation numbers 0, 1, 2, 3, 4, and so on.

Electronic circuitry can easily be designed to represent binary symbols (for example, current is flowing or not flowing) so that computer designers have chosen to use the binary number system inside of computers. Algorithms exist for translating between binary and base 10 notation, and these are incorporated into most computer systems. Thus, computer users enter their data using base 10 notation and receive their answers in base 10 notation, while the computer actually carries out computations using binary circuitry.

BIT

Binary digit.

BUG

An error in a computer program. The process of searching for bugs and correcting them is an important part of computer science and is called *debugging.*

BYTE

One character of computer memory storage. In describing a computer, one frequently gives the number of bytes of internal storage and then discusses the external storage. For example, a microcomputer may have a 48K (K=1024) internal memory and have two floppy disk drives, each using disks with a capacity of about 150,000 bytes.

CATHODE RAY TUBE (CRT)

The picture tube of a television set, and the type of display screen used on most computers. Depending on the computer system, a CRT may be limited to displaying just alphabetic and numeric characters in black and white, or it may be able to display full color graphics.

CENTRAL PROCESSING UNIT (CPU)

The part of computer hardware that fetches instructions from internal memory, interprets their meaning and carries out the instructions. The CPU of a modern, medium scale computer may be able to process ten million instructions per second.

CHARACTER

A letter, digit, punctuation mark or other symbol from a specified symbol set. A computer is designed to work with characters and with strings of characters. For example, a word is a string of one or more alphabetic characters and a number is a string of one or more numeric characters.

CHIP

A fingernail-sized circuit containing a large number of transistors and connecting components, often called an *integrated circuit.* It is called a chip because the manufacturing process makes use of a piece (chip) of silicon. The transistor was invented in 1947 and became commonplace in computers in the early 1960s. Soon people learned to manufacture a single circuit containing the equivalent of dozens or hundreds of transistors and other electronic components. The number of components in an integrated circuit has grown over the years, so that now people speak of large scale integrated circuits (LSI) and very large scale integrated circuits (VLSI). Currently, it is possible to manufacture a single chip containing the equivalent of about a half million transistors and other electronic components. Such a VLSI may serve as the CPU of a powerful microcomputer.

COMPUTER GRAPHICS

The use of a computer to deal with pictures, drawings, graphs. Computers are quite useful in commercial art and are now standardly used to help draw the cartoon shows one sees on television. Recent science fiction movies have made extensive use of computer-generated sequences. Computer graphics is of growing importance in business and industry since it provides a powerful tool for the manipulation and display of complicated data.

COBOL

COmmon Business Oriented Language, a programming language especially designed for use in business. Now more than twenty years old, this language is still the most widely used programming language for business problems. Since the language was designed strictly for use in business data processing departments, it is not a good first programming language for most students.

COMPUTER-ASSISTED LEARNING (CAL)

Any use of computers to aid in the teaching and/or learning process. Early research in CAL began in the 1950s, and now many thousands of studies have been completed. These studies have helped lead to improvements in the quality and overall usefulness of CAL. It seems likely that CAL will grow in importance and will eventually be the dominant mode of instruction in our schools.

COMPUTER LITERACY

This widely used term is not precisely defined, but refers to a level of computer knowledge and skills. Many educators suggest that all students or teachers should become computer literate. This generally means that students and teachers should gain a functional, working level of knowledge, suited to their everyday needs.

DISK

A flat, circular plate coated with magnetic iron oxide and used as an external storage device in computers. A single floppy disk may store 150,000 characters while a hard disk pack may store a half billion characters or more. A disk drive gives rapid access to programs and data stored on a disk.

HARDWARE

The physical machinery part of a computer system. It includes input, storage, manipulation and output units. The hardware for a complete computer system may range in price from a few hundred dollars to many millions of dollars.

INFORMATION RETRIEVAL (IR)

The computerized storage and retrieval of information. The external storage devices used on computers are large enough to store a dictionary, encyclopedia or large numbers of books. Eventually, these devices will store huge libraries. IR is now commonplace in business, government and research. Eventually, it will be common in schools and homes.

INTERNAL MEMORY

The fast-memory part of a computer's storage system. It contains the program that is being

run and the data as it is being processed. An inexpensive microcomputer may have an internal memory that stores a few thousand characters while the most expensive computers have internal memories that store millions of characters.

K

Computer scientists use this symbol as a measure of storage capacity, with K=1024. This is the number 2 raised to the 10th power.

LOGO

A programming language developed by Seymour Papert especially for use by young children. The language makes it particularly easy for solving certain computer graphics problems, learning some aspects of geometry and for learning some important ideas about programming.

MICROCOMPUTER

Any computer whose CPU consists of one or a small number of large scale or very large scale integrated circuits. Generally a microcomputer system is priced under $10,000, although there are no precise bounds on this.

MICROSECOND

A millionth of a second. A medium speed computer, costing perhaps a half million dollars, can perform ten multiplications in a microsecond. Such a computer can do more arithmetic in a minute than a person is apt to do by hand in a lifetime.

MODELING AND SIMULATION

A model is a representation of certain key features of an object or system being studied. Scientific models often make use of complex mathematical formulas; use of the models may require substantial amounts of computation. Computers are now a common aid to modeling. This part of computer and information science is called modeling and simulation.

NANOSECOND

A billionth of a second. The very fastest computers now under construction are approaching a speed of one operation per nanosecond. Soon we will have a computer that can count from 1 to 1,000,000,000 by 1's in a single second!

OPTICAL CHARACTER RECOGNITION

The machine reading of mark sense scan sheets, bar codes and typewritten and printed materials. This is now a common method of inputting data to a computer system.

PASCAL

A programming language especially developed for use in computer and information science courses. It is now commonly used in the first college course for students planning to major in computer and information science. Pascal is a substantially more complex programming language than BASIC or Logo.

PROCEDURE

A detailed, step-by-step plan for solving a specified type of problem. There is no guarantee that the plan will work (it may contain a bug). Computer scientists are interested in procedures that can be carried out by a computer; such procedures are called computer programs. (See also *Algorithm.*)

SOFTWARE

A detailed, step-by-step set of directions designed to be carried out by the hardware of a computer system. A computer system consists of hardware and software. In many applications the software may cost as much or more than the hardware. However, if a piece of software is widely used, its development cost can be shared among many users, so the retail price may be quite modest. Some very sophisticated computer games and/or instructional programs sell in the $20-$50 price range.

TIMESHARED COMPUTING

A form of interactive computing in which a number of terminals share the memory and central processing facilities of a computer system. This decreases the cost per user and can allow convenient communication among the users.

WORD PROCESSING

Use of a computer system as an automated typewriter. Material being typed can be viewed on a display screen, so that corrections are made before one outputs to paper. Stored material such as standard addresses, paragraphs or form letters can be used to increase the productivity of a word processor operator.

RESOURCES

MAGAZINES

Classroom Computer News, 51 Spring St., Watertown, MA 02172. ($12 for 6 issues/yr.)

Features short articles aimed at classroom teachers and administrators at the precollege level.

The Computing Teacher, University of Oregon, 1787 Agate St., Eugene, OR 97403. ($21.50 U.S., $25 Non-U.S. for 9 issues/yr. as of 10/15/83.

The Computing Teacher, journal for The International Council for Computers in Education, is designed for precollege teachers interested in the instructional use of computers. It contains material of immediate classroom use including regular columns, activities, software reviews and an annual index.

Electronic Learning, Scholastic Inc., 902 Sylvan Avenue, Box 2001, Englewood Cliffs, NJ 07632. ($19.00 for 8 issues/1 yr.; introductory offer $15.95.)

Published by Scholastic, Inc., *Electronic Learning* is aimed at beginning users of calculators, computers and other electronic aids to learning.

There are numerous other general and specialized journals and magazines that deal with microcomputing. Browse the rack in your local computer store to find your favorite.

BOOKS

Harper, Dennis O. and Stewart, James H., Editors. *Run: Computer Education.* Brook/Cole Publishing Co., 1983, $15.95.

This book is an extensive collection of articles and papers which deal with many of the facets of computers in the educational process.

Moursund, David. *Introduction to Computers in Education for Elementary and Middle School Teachers.* International Council for Computers in Education, University of Oregon, 1787 Agate St., Eugene, OR 97403, 1981, $7.00.

This book is designed for self-instruction or for a formal inservice or preservice course for teachers.

Papert, Seymour. *Mindstorms: Children, Computers and Powerful Ideas.* Basic Books, 1980.

This book presents a forward-looking view of the learning that can go on when children, computers and the Logo language are combined.

Taylor, Robert, Editor. *The Computer in the School: Tutor, Tool, Tutee.* Teachers College Press, 1980.

This collection features papers from five leading thinkers in educational computing: Alfred Bork, Thomas Dwyer, Arthur Luehrmann, Seymour Papert and Patrick Suppes.

ORGANIZATIONS

ICCE (International Council for Computers in Education), University of Oregon, 1787 Agate St., Eugene, OR 97403. (503) 686-4414.

ICCE is a professional, non-profit organization for people interested in instructional computing at the precollege level. It has more than 12,000 Individual Members and 39 Organization Members made up of regional, state or province-wide and special interest groups. ICCE publishes a number of booklets of interest to computer-using educators as well as *The Computing Teacher* journal.

MECC (Minnesota Educational Computing Consortium), 2520 Broadway Dr., St. Paul, MN 55113. (612) 376-1101.

MECC provides services for students, teachers and administrators in Minnesota public schools and colleges. MECC offers inservice training and curriculum guides, develops and distributes educational software and is a source of software and materials written for Apples and Atari. (Contact MECC for charges to non-residents of Minnesota.)

ICCE PUBLICATIONS

• The **School Administrator's Introduction to Instructional Use of Computers** gives a good fundamental overview for school administrators and board members. It maps out the basic elements of computer education and ways that computers are affecting today's curriculum. The booklet introduces the subject, reviews instructional uses of computers, looks at the problems and processes of goal development, and closes with a glossary and brief resource guide. 5½x8½, 64 pages, revised April 1983.

• The **Teacher's Guide to Computers in the Elementary School** is designed for the teacher with little practical computer experience. The booklet presents an informative introduction to computers, discusses the use of computers in the elementary classroom, and suggests activities, many of which do not require a computer. A glossary and list of additional resources is included. 5½x8½, 48 pages.

• **Precollege Computer Literacy: A Personal Computing Approach** encourages educators to develop and implement a universal standard of pre-college computer literacy. The booklet begins by presenting background information on computer technology, educational uses of computers and computer literacy. It discusses five areas where computers can be personally useful to students and how each contributes to a definition of computer literacy. A glossary is also included. 5½x8½, 28 pages.

• An **Introduction to Computers in Education for Elementary and Middle School Teachers** is designed for self-instruction, inservice or preservice use and is suitable for teachers at all levels. This in-depth book includes eight chapters with over 75 applications and exercises (many of which do not require computer access) that can be used at the elementary and middle school levels. 8½x11, 94 pages.

• The **Parent's Guide to Computers in Education** is an excellent resource for parents who want to become actively involved in their children's computer education. The booklet talks about why children should use and learn about computers and what parents can do to help. There is a section on how students and teachers are using computers in the classoom and suggestions on how families can get hands-on experience. Checklists help parents who plan to purchase a home computer. A glossary and list of resources are included. To put the subject into perspective, a light-hearted piece by Merle Marsh, "Comes the Dawn (If Only I Can Find the Switch)," runs throughout the booklet. 5½x8½, 80 pages.

• **An Introduction to Computing: Content for a High School Course** proposes a unit-by-unit course outline for computer science instruction at the secondary school level. Recently revised, and designed for a year-long course, the booklet contains a variety of activities emphasizing applications, programming, computer environment and social impact. 5½x8½, 48 pages.

• **The Evaluator's Guide for Microcomputer-Based Instructional Packages** can help you make informed, intelligent software purchasing decisions. This booklet is a step-by-step guide to concise software evaluation. It includes sample reviews, evaluation forms and a glossary and is an invaluable tool in effective software review and selection. 5½x8½, 48 pages.

• **Computers in Composition Instruction** includes the proceedings of a research/practice conference held at SWRL Education Research and Development in 1982. The booklet talks about the courseware available to assist in language arts instruction, the current level of computer-assisted composition instruction and examines the use of computers at the prewriting and composition stages. Many programs are discussed and an annotated bibliography and list of resources is included. Edited by Dr. Robert Shostak, Florida International University. Available winter 1983. 5½x8½, 88 pages.

• **Computer Literacy Activities for Elementary and Middle School Students** is a collection of twelve introductory activities for computer literacy concepts that any teacher—even those just getting started—can understand and use. Designed especially for elementary and middle school teachers, this book contains all the background knowledge necessary for the lessons and is easily integrated into the curriculum. Most activities do not require computer access. 8½x11, 56 pages.

• **Learning Disabled Students and Computers: A Teacher's Guide Book** was written to help those teachers who are responsible for meeting the needs of children legally identified as learning disabled. The booklet discusses software and hardware and emphasizes educational computing theory, concerns and a vision of computers in education for learning disabled students. The resource section contains a glossary and references to relevant books, publishers and organizations. 5½x8½, 48 pages.

• **Computer Technology for the Handicapped in Special Education & Rehabilitation: A Resource Guide** is a detailed, up-to-date annotated bibliography of resources in the area of computers for the handicapped in rehabilitation and special education. Containing an author index, subject index and referenced quotations, the focus of the guide is a body of nearly 200 detailed annotations of current writings in the field. Specific subject areas include communication, computer assisted instruction, disability/handicap, functional aids, microcomputers/applications, newsletter/bulletins, periodicals, rehabilitation, special education, teachers/service providers, universities/colleges and several others. 8½x11, 94 pages.

• **Microcomputers in the Classroom—Dreams and Realities** is directed to educators, administrators, curriculum committees and other professionals concerned about the practice and potential of computer education. Written by Dr. Henry Jay Becker of The Johns Hopkins University, this booklet provides an overview of the existing computer education environment as it relates to school systems and policy. Dr. Becker has included a summary of current theory, a review of relevant research and an outline of some major factors that will influence purchasing, implementation and goal-setting decisions. Divided into fourteen sections such as "management of instruction" and "curriculum-modifying applications of microcomputers," the booklet concludes with suggestions for improved performance on the part of researchers, developers and school systems. 5½x8½, 64 pages.

• **Computer Metaphors: Approaches to Computer Literacy** introduces a new way of thinking for beginners and advanced computer users by relating a computer to more familiar concepts. Useful as a basis for classroom discussions, this package develops the idea of the computer as brain, as person, as glass box, as palette and five additional metaphors. Developed by Dr. Howard Peelle, University of Massachusetts.

Package includes an illustrated 5½x8½ booklet and a poster-size drawing of each of the nine metaphors.

• **Logo in the Classroom** takes Logo a step beyond "playing turtle" and integrates it into the elementary curriculum. The twenty-three Logo lessons were conceived and developed in a classroom setting and are designed to meet the needs of an entire class at one time! Guidelines are included to help the elementary teacher when computers are in short supply. Logo dialects covered are MIT Logo (Terrapin & Krell) for Apple and Commodore, LCSI for Apple, Atari Logo, Texas Instruments Logo and Radio Shack Logo. Written by Shirley R. Torgerson, Mary Kay Kriley and Janet T. Stone. Available winter 1983; 8½x11.

• **Topics: Computer Education for Colleges of Education** contains detailed descriptions of computer education programs currently operating at sixteen North American colleges of education. The wide variety of programs included identifies alternatives for developing computers in education programs. Published by the Association for Computing Machinery and edited by Jean Rogers. 8½x11, 111 pages.

To purchase these booklets write to ICCE, Dept. CTA, 1787 Agate St., University of Oregon, Eugene, OR 97403 or phone 503/686-4414.

ICCE Organization Members

Alaska Association for Computers in Education (AACE)
Contact: Sandra Burgess
Coordinator, The Northern Institute, 650 W. International Airport Rd., Anchorage, AK 99502

Alberta Association for Educational Data Systems
Contact: Keath Austen
838 Education Tower, University of Calgary, 2500 University Dr. NW, Calgary, Alberta, Canada T2N 1N4

Arizona Computer Users in Education
Contact: Nancy Watson
2148 Santos Drive
Tempe, AZ 85281

Association for the Development of Computer-Based Instructional Systems (ADCIS)
Contact: Dr. Gordon Hayes, Executive Director
ADCIS—Miller Hall 409, Western Washington University, Bellingham, WA 98225

Computer Education Group of Victoria
Contact: Leon Guss, Conference Secretary
P.O. Box 245, Niddrie 3042,
Melbourne, Victoria, Australia

Computer Education Group of Queensland
Contact: D.A. Carey
G.P.O. Box 1669
Brisbane, Queensland 4001
Australia

Computer Education Society of Ireland
Contact: Mary Devlin
85 Limetrees Road East, Douglas, Cork, Ireland

Computer Educators of Idaho
Contact: Dennis Dempsey
P.O. Box 1088, Hailey, ID 83333

Computers, Learners, Users, Educators Association (CLUES)
Contact: Henry J. Petersen, Executive Director
50 Nellis Drive, Wayne, NJ 07470

Computer-Using Educators (California)
Contact: Ms. Bobby Goodson
P.O. Box 18547, San Jose, CA 95158

Computer-Using Educators of British Columbia (CUEBC)
Contact: Barry Underwood
509 Alder St., Campbell River,
B.C., Canada V9W 2N9

Computer-Using Educators of Kentucky (CUE-KY)
Contact: Janet Parker
EMCE Dept., School of Education,
University of Louisville, Louisville, KY 40292

DIDACOM
Contact: Inno Broekman
Avenbeeck 98, 2182 RZ Hillegom,
The Netherlands

Educational Computing Consortium of Ohio
Contact: Ellen Richman, Coordinator
4777 Farnhurst Road, Cleveland, OH 44124

Educational Computing Association of Western Australia
Contact: P.C. Farrell, President
12 Lilac Place, Dianella, 6062, Western Australia

Educational Computing Organization of Ontario (ECOO)
Contact: Robert E. Drake, President
252 Bloor Street West, Toronto,
Ontario, Canada M5S 1V6

Educational Microcomputer Users Group of Central New York (EMUGCNY)
Contact: Jane McCrohan
Roxboro Middle School, Mattydale, NY 13211

Educators Interest Group of the San Diego Computer Society
Contact: Melvin L. Zeddies
P.O. Box 81537, San Diego, CA 92138

Florida Association of Science Teachers (FAST)
Contact: Richard Sweetsir
1147 Scotten Rd., Jacksonville, FL 32205

Illinois Association for Educational Data Systems (ILAEDS)
Contact: Lyle B. Smith, President
Computer Science Department, Northern Illinois University, Box 128, De Kalb, IL 60115

Indiana Computer Educators
Contact: David A. Flowers
Ft. Wayne Community Schools, Adm., Cwt.
1230 So. Clinton St., Ft. Wayne, IN 46802

Manitoba Association for Educational Data Systems (MAN-AEDS)
Contact: Ed Boorsma, Secretary-Treasurer
1577 Wall Street, East Winnipeg, Manitoba,
Canada R3E 2S5

Michigan Association for Computer Users in Learning (MACUL)
Contact: Lary R. Smith, Communications Sect.
33500 Van Born Road, Wayne, MI 48184

Minnesota Association for Educational Data Systems
Contact: Sue Talley
1925 W. County Rd. B2, St. Paul, MN 55113

Montana Council for Computers in Education
Contact: Leon Roland
1846 Bitteroot
Billings, MT 59105

National Institute for Microcomputer Based Learning (NIMBL)
Contact: Stanley Silverman, President
348 Plymouth Ave., Brightwaters, NY 11718

New Hampshire Association for Computer Education Statewide (NHACES)
Contact: Anne Knight, President
University of New Hampshire, Computer Services, Durham, NH 03824

New York State Association for Educational Data Systems
Contact: Don Ross, President
Ardsley High School, 300 Farm Road,
Ardsley, NY 10502

New South Wales Computer Education Group
Contact: Toni Downes, Chairperson
P.O. Box 148, Broadway, N.S.W. 2007 Australia

Northwest Council for Computer Education (Oregon, Washington and Northern Idaho)
Contact: Howard Bailey
Computing Center, Eastern Oregon State College
La Grande, OR 97850

Oklahoma Educational Computer Users Program (OECUP)
Contact: Richard V. Andree
Department of Mathematics, 601 Elm St., #423,
The University of Oklahoma, Norman, OK 73019

Pennsylvania Learning Resources Association (PLRA)
Contact: George H. Zook, Director, IMS
Lancaster-Lebanon I.U./IMS
P.O. Box 5026, Lancaster, PA 17601

Quebec Association of Computer Users in Education
Contact: Don Rosenbaum, Membership Director
P.O. Box 369, Ormstown, Quebec, J0S 1K0,
Canada

Saskatchewan Association for Computers in Education
Contact: Duncan Campbell
Mt. Royal Collegiate, 2220 Rusholme Road,
Saskatoon, Saskatchewan, Canada S7L 4A4

The Science Teachers' Association of Ontario
Contact: Warren Sirrs
1032 Harkness Ave., Ottawa, Ontario,
Canada K1V 6P2

Society of Data Educators (SDE)
Contact: Robert Behling
Bryant College, Smithfield, RI 02917

Society for Microcomputers in Life & Education
Contact: Hillel Weintraub, Acting President
Doshisha International High School
Tatara, Tanabe-cho, Tsuzuki-gun
Kyoto-fu, Japan 610-03

Texas Computer Education Association
Contact: Vicki S. Smith
P.O. Box 2573, Austin, TX 78768

The Utah Council for Computers in Education
Contact: Dr. Larry C. Christensen
1295 North 1200 West, Mapleton, UT 84663

Wyoming Educational Computing Council
Contact: Robert L. Morissette, President
Laramie County School District #1, Admin. Bldg.
2810 House Ave., Cheyenne, WY 82001

Young People's LOGO Association (YPLA)
Contact: James H. Muller
1208 Hillsdale Drive, Richardson, TX 75081